



图灵计算机科学丛书



C语言详解 (第5版)

Problem Solving and Program Design in C
Fifth Edition

[美] Jeri R. Hanly Elliot B. Koffman 著
万波 潘蓉 郑海红 译



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵计算机科学丛书

C语言详解

(第5版)

Problem Solving and Program Design in C

Fifth Edition

[美] Jeri R. Hanly Elliot B. Koffman 著
万波 潘蓉 郑海红 译



人民邮电出版社
北京

图书在版编目(CIP)数据

C语言详解:第5版/(美)汉利(Hanly, J.R.), (美)科夫曼(Koffman, E.B.)著;万波,潘蓉,郑海红译.
—北京:人民邮电出版社,2007.11
(图灵计算机科学丛书)
ISBN 978-7-115-16427-8

I. C… II. ①汉…②科…③万… III. C语言—程序设计 IV. TP312

中国版本图书馆CIP数据核字(2007)第090594号

内 容 提 要

本书是C语言的经典教材。与同类图书相比,本书的最大特色在于,不仅深入浅出地讲述了实际开发最需要的C语言基础知识以及动态数据结构和多进程等高级内容,而且在此过程中同时阐述程序设计思想,注重提高学生的问题解决能力和实际编程能力,使学生能够了解和初步掌握当前软件行业公认的程序设计风格和编程实践。书中对指针的处理非常合理,有效地降低了这一难点的学习门槛。此外,本书有丰富的教学辅助内容,配有各种层次的习题和示例,而且每一章都有一个或多个来自实际生活的实例研究。

本书可以作为计算机专业及其他专业本科生C语言程序设计课程的教材;对已有C语言编程经验的技术人员,本书也是不可多得的参考书。

图灵计算机科学丛书

C语言详解(第5版)

- ◆ 著 [美]Jeri R. Hanly Elliot B. Koffman
译 万波 潘蓉 郑海红
责任编辑 杨海玲
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京顺义振华印刷厂印刷
新华书店总店北京发行所经销
- ◆ 开本: 787×1092 1/16
印张: 34.75
字数: 1 028 千字 2007年11月第1版
印数: 1—4 000 册 2007年11月北京第1次印刷
著作权合同登记号 图字: 01-2007-0540 号

ISBN 978-7-115-16427-8/TP

定价: 69.00 元

读者服务热线: (010)88593802 印装质量热线: (010)67129223

版 权 声 明

Authorized translation from the English language edition, entitled *Problem Solving and Program Design in C, Fifth Edition*, 0321409914 by Jeri R. Hanly and Elliot B. Koffman, published by Pearson Education, Inc., publishing as Addison Wesley, Copyright © 2007 by Pearson Education, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

CHINESE SIMPLIFIED language edition published by PEARSON EDUCATION ASIA LTD. and POSTS & TELECOM PRESS Copyright © 2007.

本书中文简体字版由Pearson Education Asia Ltd.授权人民邮电出版社独家出版。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。

本书封面贴有Pearson Education（培生教育出版集团）激光防伪标签，无标签者不得销售。

版权所有，侵权必究。

译者简介

- 万 波** 西安电子科技大学计算机学院讲师、在职博士。研究方向为计算机系统结构、图形图像处理，曾参与多项武器装备预研基金、陕西省自然科学基金等项目。目前为计算机学院程序设计课程组组长，承担的教学课程有图形学、Java程序设计、C++程序设计、网络应用程序设计等，其中C++程序设计是陕西省精品课程。
- 潘 蓉** 西安电子科技大学计算机学院讲师、博士。科研方向为图形图像处理、数字水印技术等，主要教授面向对象程序设计和网络应用程序设计等课程，在核心期刊及会议上已发表10余篇论文。
- 郑海红** 西安电子科技大学计算机学院讲师、博士。研究方向为计算机图形图像处理，曾参与武器装备预研基金、陕西省自然科学基金等项目，并在核心期刊、国际会议上发表论文10余篇。承担的教学课程有离散数学、人工智能等。

译者序

随着面向对象编程和可视化集成开发工具的发展，很多曾经非常流行的编程语言的影响力正在下降，甚至逐渐消失，但有一种语言例外，它就是C语言！时光的流逝丝毫没有减少C的魅力，它的风采依然。

因为C语言所涉及的概念比较复杂，规则繁多，用法灵活，容易出错，所以很多初学者感到无从下手。初学者通常很难理解C语言的某些概念，虽然非常渴望学好C语言，但往往因为难以入门而望而却步。更为重要的是，很多学生虽然掌握了C语言的每个知识点，但在解决实际问题时，却仍然不知所措，而本书的宗旨正是强调如何利用C语言解决问题，相信读完本书之后，读者在解决问题方面一定会有豁然开朗的感觉。

除了语言生动之外，本书还具备许多其他书籍没有的特色。

- 在介绍新的概念之前，通常先引入一个贴切有趣的应用示例，有助于调动初学者的兴趣并顺利实现初学者对概念的理解。
- 程序设计思想的介绍和C语言语法的讲述双管齐下，使学生先理解程序设计的宏观概念，然后在此基础上学习C语言的具体语法，有助于加深其对C语言的理解，增强其驾驭C语言的能力，在实际应用中做到举一反三，灵活自如。
- 结构图的采用使得本书图文并茂；例程代码清晰完整，为学生仔细研读程序提供了可能。
- 每章都有一个来自于具体应用的完整的实例研究，虽然由于国情不同，对某些应用的细节国内读者可能并不熟悉，但这些实例的讲述却能够使读者在贴近现实的环境中应用所学到的C语言知识，这也是本书的亮点之一。
- 各章均点出了常出现的错误，为初学者提供了宝贵的实践经验。
- 书中很多地方涉及程序风格，这一点是国内书籍涉及甚少的一方面，根据译者的教学经验，这一点相当重要。国内学生在这方面欠缺甚多，通常都认为程序风格不如编写程序重要，然而良好的程序风格将会使每一个从事程序设计的人受益无穷，对于学习程序设计的学生更显得尤为重要。

本书是为计算机专业低年级本科生以及众多其他学科的初学者或自学者编写的C语言课程教材，也可供那些已有一些C语言基础但还想进一步提高的读者参考。

参加本书翻译工作的人员有万波、潘蓉、郑海红等同志。此外特别感谢我的妻子孟广兰在英语语言方面提供的帮助。衷心感谢人民邮电出版社图灵公司的编辑所做的大量细致的工作。

由于时间仓促，译者水平有限，译文中难免还存在不少欠妥和纰漏之处，恳请广大读者不吝赐教和指正。

万 波

2007年3月 于西安电子科技大学

前言

本书教给读者怎样用训练有素的方法来解决问题，并应用广为接受的软件工程方法将程序解决方案设计为内聚的、可读的、可重用的模块。本书将ANSI C的一个子集作为这些模块的实现工具进行介绍，ANSI C是一种标准化的、具有工业强度的编程语言，功能强大，可移植性好。本书可以用于程序设计方法的第一门课程，因为它不要求计算机和程序设计的预备知识；此外，本书选择的大量实例研究和练习使得教师可以用本书为计算机专业或其他专业的学生开设一门介绍性的C语言程序设计课程。

本版的更新

第15章“使用进程和线程的多进程”是全新的一章，讲授了多进程基本概念。这一章向学生介绍了理解事件驱动和中断驱动编程所必需的概念，包括父进程和子进程以及进程间通信、线程、互斥锁机制和避免死锁的技术。这一章教会学生如何使用C函数来支持这些概念。

用 C 来教授程序开发

本书的两个目标（讲授程序设计以及讲授C）在某些人看来可能是矛盾的。很多人认为C语言应该在学习过用其他更加友好的语言进行编程之后再学习。C语言之所以被认为非常困难，可以追溯到它的历史。C语言是作为在UNIX操作系统下编程的工具而设计的，它的最初的使用者就是理解操作系统和底层机器复杂性的程序员，这些使用者认为在其程序中利用这些知识很自然。所以，司空见惯的是，许多讲授C语言的教材展示给学生的程序示例都需要理解机器概念，尽管这些概念并没有出现在标准的初级程序设计课程的大纲中。

使用本书既能够讲授程序开发的正确方法又可以讲授ANSI C语言，而且本书选择前者作为最主要的目标。有人可能会担心这种选择会弱化对ANSI C的阐述。事实正相反，我们发现将程序设计的概念和这些概念在C中的实现相结合，能够清晰地展示C作为一种高级编程语言所具备的威力，而这一点在那些以介绍所有ANSI C内容为主的教科书中往往是很模糊的。虽然与讨论C语言特征相比我们更强调程序设计的方法，但本书仍然全面讲述了C的基本结构。

指针与本书的组织

书中C语言主题出现的顺序由初级编程者的需要来决定，而不是由C语言的结构来决定。读者会惊奇地发现没有标题为“指针”的章节。这是因为我们将C当作一种高级语言来讨论，而并不是忽视了指针在C语言中的重要作用。

尽管其他高级语言在输出参数和数组方面具有独立的语言结构，但C语言将这些概念包装在它的指针概念中，这明显增加了学习这种语言的难度。本书在相应的位置分散地讨论指针，从而简化了学习过程，这使得学生能够每次吸收一点指针用法的精髓。本书的方法使得用传统的高级语言术语（输出参数、数组、数组下标和字符串）来表示基本概念成为可能，并使得没有汇编语言背景的学生掌握指针的多种用法变得更加容易。

因此，本书不是用1章而是用4章来强调指针。第6章讨论了指针作为输出和输入/输出参数的用法，第8章讨论了数组，第9章讲述了字符串和指针数组，第14章在回顾了以前讨论的指针用法后描述了动态内存分配。此外，第2章和第12章讨论了文件指针。

软件工程概念

本书涉及了软件工程的许多方面，其中有一些是明确讨论的，而另外一些则是通过例子给出的。良好的问题解决技巧和有效的软件开发之间的联系在第1章讨论问题解决的科学和艺术时就已经建立。第1章给出的5步软件开发方法用于解决第一个实例研究，并贯彻于本书的其他实例研究中。重要的程序风格用特定方式突出显示，而且示例中使用的编码风格符合软件行业的规范要求。有几章的小节还讨论了算法跟踪、程序调试和测试。

第3章通过介绍部分C库函数、无参数void函数以及接受输入参数并返回值的函数介绍了过程抽象。第4章和第5章包括了更多的函数示例，第6章完成了具有简单参数的函数研究，还讨论了使用指针表示输出和输入/输出参数，而且第7章介绍了怎样把函数作为参数使用。

第6章、第8章和第11章的实例研究和示例程序介绍了数据抽象的概念以及数据类型和运算符封装的概念。第13章给出了在个人库中规范过程和数据抽象的C工具，这些个人库由单独的头文件和实现文件定义。第16章介绍了C++实现的面向对象设计概念。

整本书中强调了可视函数接口的使用。直到第13章才提到使用全局变量的可能性，然后仔细地描述了使用全局变量的优点和危害。

教学辅助内容

本书使用了以下教学辅助内容来增强本书作为学习工具的实用性。

- **小节末尾的练习。**大部分小节结尾都有自测练习，分为需要分析程序段的练习和简短的编程练习。本书还配有部分自测练习的答案^①。
- **示例和实例研究。**本书包含了各种各样的编程示例。只要可能，示例就给出了完整的程序或函数而不是未完成的程序片段。每一章都包含了一个或多个具有实际意义的实例研究，而且都遵循软件开发方法来解决。大量实例研究让学生看到了计算的重要应用，包括数据库查询、用于记账和销售分析的商业应用、文字处理以及诸如辐射等级监测和节约用水等有关环境的应用。
- **语法展示框。**语法展示框描述了语法和新的C特性的语义并给出了例子。
- **程序风格展示。**程序风格展示并讨论了良好的编程风格。
- **错误讨论和每章复习。**每一章都有一小节讨论常见编程错误。每章复习经常以表格形式给出本章介绍的新的C语言结构。
- **章末练习。**每一章复习之后是附有答案的快速测试练习。同时还有复习题。

^① 可以从图灵网站（www.turingbook.com）下载。——编者注

- 章末项目。每一章末尾是程序设计项目。

附录

附录A给出了字符集表。由于本书只讨论了ANSI C的一个子集，其余的附录对提升本书作为参考书的价值非常重要。附录B是ANSI C的标准库函数的表格，它们按字母排序。附录C中给出了所有C运算符的优先级和结合性的表格，而且给出了前面没有讨论过的运算符的解释。整本书中数组引用都通过下标进行。附录D讨论了指针运算。附录E列出了所有的C保留字。ANSI C结构的参照表在本书附录F中给出。^①

补充

本书的所有读者都可以从www.aw.com/cssupport得到以下补充材料^②：

- 源代码。
- 已知的勘误表。

致谢

许多人参与了这本书的出版过程。我们感谢WaveRules公司的Jeff Warsaw，他对本书的新章节做出了重要贡献。加利福尼亚理工大学喷气推动实验室的Joan C.Horvath提供了几个程序设计练习。我们也感激那些在过去几年中帮助验证编程示例，并提供练习答案的Temple大学和怀俄明大学的学生，包括Mark Thoney、Lynne Doherty、Andrew Wrobel、Steve Babiak、Donna Chrupcala、Masoud Kermani、Thayne Routh和Paul Onakoya。

在此过程中很高兴和Addison-Wesley团队一起合作。策划编辑Michael Hirsch和编辑助理Lindsey Triebel在原稿修订的各个阶段提供了指导和鼓励。Marilyn Lloyd负责了本书的生产，而Michelle Brown和Dana Lopreato负责了市场运作。

作者

① 所有附录均可从图灵网站（www.turingbook.com）下载。——编者注

② 亦可从图灵网站下载。已知勘误已在中译本中更正。——编者注

目 录

第 1 章 计算机与程序设计概述	1	2.2.1 变量声明	24
1.1 电子计算机的过去和现在	1	2.2.2 数据类型	25
1.2 计算机硬件	2	2.3 可执行语句	27
1.2.1 存储器	3	2.3.1 内存中的程序	27
1.2.2 中央处理器	6	2.3.2 赋值语句	27
1.2.3 输入/输出设备	6	2.3.3 输入/输出操作和函数	28
1.2.4 计算机网络	7	2.3.4 printf函数	29
1.3 计算机软件	9	2.3.5 scanf函数	31
1.3.1 操作系统	9	2.3.6 return语句	32
1.3.2 应用软件	10	2.4 C程序的一般形式	33
1.3.3 计算机语言	10	2.4.1 程序风格: 程序中的空格	33
1.3.4 执行程序	13	2.4.2 程序的注释	34
1.4 软件开发方法	14	2.4.3 程序风格: 使用注释	34
1.4.1 软件开发方法	14	2.5 算术表达式	35
1.4.2 注意: 失败是编程过程的一部分	16	2.5.1 运算符/和%	36
1.5 应用软件开发方法	16	2.5.2 表达式的数据类型	37
实例研究: 英里到公里的转换	16	2.5.3 混合类型赋值语句	37
复习	18	2.5.4 强制类型转换	38
快速测试	19	2.5.5 具有多个运算符的表达式	38
快速测试答案	19	2.5.6 用C语言编写数学公式	40
复习题	19	实例研究: 对收集的硬币求值	41
第 2 章 C 概述	20	2.6 在程序输出中格式化数值	44
2.1 C语言要素	20	2.6.1 格式化int型值	44
2.1.1 预处理指令	20	2.6.2 格式化double型值	45
2.1.2 预处理指令的语法展示	21	2.6.3 程序风格: 消除前导空白	46
2.1.3 main函数	22	2.7 交互模式、批处理模式和数据文件	46
2.1.4 保留字	22	2.7.1 输入重定向	46
2.1.5 标准标识符	23	2.7.2 程序风格: 回显与提示符	47
2.1.6 用户定义的标识符	23	2.7.3 输出重定向	47
2.1.7 大写和小写字母	24	2.7.4 程序控制的输入和输出文件	48
2.1.8 程序风格: 选择标识符名称	24	2.8 常见编程错误	49
2.2 变量声明和数据类型	24	2.8.1 语法错误	49
		2.8.2 运行时错误	51
		2.8.3 未检测到的错误	51
		2.8.4 逻辑错误	53

复习	53	第4章 选择结构: if 语句和 switch	
快速测试	54	语句	88
快速测试答案	55	4.1 控制结构	88
复习题	55	4.2 条件	88
程序设计项目	56	4.2.1 关系运算符和判等运算符	89
第3章 函数的自顶向下设计	58	4.2.2 逻辑运算符	89
3.1 利用已有信息编程	58	4.2.3 运算符优先级	90
实例研究: 求圆的面积和周长	59	4.2.4 短路求值法	92
实例研究: 计算一批平整圈的重量	61	4.2.5 用C语言表示条件	92
3.2 库函数	64	4.2.6 比较字符	93
3.2.1 预定义函数和代码复用	64	4.2.7 逻辑赋值	93
3.2.2 使用灰色区分新结构	65	4.2.8 条件取反	94
3.2.3 C库函数	65	4.2.9 德摩根定理	95
3.2.4 本节目标	67	4.3 if语句	96
3.3 自顶向下设计和结构图	68	4.3.1 有两个选项的if语句	96
实例研究: 画简单的图	68	4.3.2 只有一个选项的if语句	96
3.4 无参函数	69	4.3.3 比较一个选项的if语句和两个选项的if语句	97
3.4.1 函数原型	70	4.3.4 程序风格: if语句格式	98
3.4.2 函数定义	70	4.4 具有复合语句的if语句	98
3.4.3 程序中函数的位置	72	4.4.1 程序风格: 编写具有复合语句的if语句	99
3.4.4 程序风格: 在含有函数的程序中使用注释	73	4.4.2 跟踪if语句	99
3.4.5 子函数和主函数的执行顺序	73	4.5 算法中的决策步骤	101
3.4.6 使用子函数的优点	74	实例研究: 水费问题	101
3.4.7 显示用户指令	74	4.5.1 程序风格: 在函数中使用一致的名称	106
3.5 带输入参数的函数	75	4.5.2 程序风格: 内聚函数	106
3.5.1 带输入参数的void函数	76	4.5.3 程序风格: 使用常量宏来增强可读性和易维护性	107
3.5.2 带输入参数和单一结果的函数	77	4.6 解决更多问题	107
3.5.3 程序风格: 函数接口注释	79	4.6.1 结构图中的数据流信息	107
3.5.4 带多个参数的函数	79	4.6.2 使用子函数修改程序	108
3.5.5 参数表一致性	80	实例研究: 具有节约需求的水费问题	108
3.5.6 函数数据域	81	4.7 嵌套if语句和多选项决策	109
3.5.7 使用驱动测试函数	81	4.7.1 比较嵌套if和if序列	110
3.6 常见编程错误	82	4.7.2 嵌套if的多选项决策形式	110
复习	82	4.7.3 多选项决策中条件的顺序	111
快速测试	83	4.7.4 程序风格: 确认变量的值	113
快速测试答案	84	4.7.5 具有多个变量的嵌套if语句	113
复习题	84	4.8 switch语句	116
程序设计项目	84		

4.9 常见编程错误.....	119	第6章 模块化编程.....	171
复习	120	6.1 带简单输出参数的函数	171
快速测试	121	6.2 带输入输出参数函数的多次调用	176
快速测试答案.....	123	程序风格: 首选的函数类型	179
复习题	123	6.3 名字的作用域	180
程序设计项目	124	6.4 输出形参作为实参变量	181
第5章 重复和循环语句.....	128	6.5 由多个函数构成的程序	184
5.1 程序中的重复	128	实例研究: 普通分数的算术运算	184
5.2 计数循环和while语句	129	6.6 程序系统的调试与测试	190
5.3 在循环中计算和或者乘积	131	6.7 常见编程错误	192
5.3.1 程序风格: 编写通用循环	133	复习	192
5.3.2 计算一列数的乘积	133	快速测试	193
5.3.3 复合赋值运算符	134	快速测试答案	194
5.4 for语句	135	复习题	194
5.4.1 程序风格: 格式化for语句	136	程序设计项目	195
5.4.2 递增和递减运算符	136	第7章 简单数据类型	201
5.4.3 1之外的递增和递减	138	7.1 数值类型的表示与转换	201
5.4.4 显示值构成的表格	139	7.1.1 数值类型的差异	201
5.5 条件循环	140	7.1.2 数值的不精确性	203
程序风格: 在子函数中执行循环处理 ..	144	7.1.3 数据类型的自动转换	203
5.6 循环设计	144	7.1.4 数据类型的显式转换	204
5.6.1 标记控制循环	145	7.2 字符类型的表示与转换	205
5.6.2 使用for语句实现标记循环	146	7.3 枚举类型	206
5.6.3 文件结束控制循环	147	7.4 迭代近似	211
5.6.4 错误数据上的无限循环	148	实例研究: 求根的二分方法	212
5.7 嵌套循环	149	7.5 常见编程错误	217
5.8 do-while语句和标志控制循环	152	复习	217
5.9 问题求解示例	154	快速测试	218
实例研究: 太阳能加热房的采集面积 ..	154	快速测试答案	219
5.10 如何调试并测试程序	159	复习题	219
5.10.1 使用调试器程序	159	程序设计项目	220
5.10.2 不使用调试器进行调试	159	第8章 数组	225
5.10.3 循环次数差1的错误	160	8.1 声明和引用数组	225
5.10.4 测试	160	8.2 数组下标	228
5.11 常见编程错误	161	8.3 使用循环顺序存取	229
复习	162	8.3.1 使用数组进行统计计算	230
快速测试	164	8.3.2 程序风格: 使用循环控制变量 ..	232
快速测试答案	165	作为数组下标	232
复习题	165	8.4 将数组元素作为函数参数使用	232
程序设计项目	166		

8.5 数组参数	234	9.8 字符串处理示例	290
8.5.1 形参数组	234	实例研究: 文本编辑器	291
8.5.2 和形参数组对应的实参	235	9.9 常见编程错误	297
8.5.3 数组作为输入参数	236	复习	298
8.5.4 返回数组结果	237	快速测试	299
8.5.5 部分填充数组	239	快速测试答案	300
8.5.6 栈	240	复习题	301
8.6 数组搜索和排序	243	程序设计项目	302
8.6.1 数组搜索	243	第 10 章 递归	305
8.6.2 数组排序	244	10.1 递归的本质	305
8.7 多维数组	246	10.2 跟踪递归函数	308
8.7.1 多维数组的初始化	248	10.2.1 带返回值的递归函数的跟踪	308
8.7.2 多维数组	248	10.2.2 void类型递归函数的跟踪	309
8.8 数组处理示例	249	10.2.3 参数与局部变量栈	311
实例研究: 销售数据分析	249	10.2.4 C中参数栈的实现	312
8.9 常见编程错误	255	10.2.5 何时、如何跟踪递归函数	313
复习	256	10.3 递归数学函数	314
快速测试	257	10.4 带数组和字符串参数的递归函数	318
快速测试答案	257	实例研究: 找出字符串中的大写字母	318
复习题	258	实例研究: 递归选择排序	320
程序设计项目	258	10.5 用递归解决问题	322
第 9 章 字符串	264	实例研究: 集合运算	322
9.1 字符串基础	264	10.6 递归的经典实例研究: 汉诺塔	327
9.1.1 声明并初始化字符串变量	264	10.7 常见编程错误	331
9.1.2 字符串数组	265	复习	332
9.1.3 使用printf和scanf进行输入/ 输出	265	快速测试	332
9.2 字符串库函数: 赋值和子串	268	快速测试答案	333
9.2.1 字符串赋值	268	复习题	333
9.2.2 子串	270	程序设计项目	334
9.3 较长的字符串: 拼接和整行输入	274	第 11 章 结构体与共用体类型	337
9.3.1 拼接	274	11.1 用户自定义结构体类型	337
9.3.2 字符和字符串的区别	275	11.1.1 结构体类型定义	337
9.3.3 输入一个完整的行	275	11.1.2 操作结构化数据对象的单个 成员	339
9.4 字符串比较	277	11.1.3 运算符优先级回顾	339
9.5 指针数组	279	11.1.4 操作结构体整体	340
9.6 字符操作	283	11.1.5 程序风格: 类型命名约定	340
9.6.1 字符输入/输出	283	11.2 结构体类型数据作为输入/输出参数	341
9.6.2 字符分析和转换	284	11.3 返回值为结构体类型的函数	345
9.7 字符串到数值和数值到字符串的 转换	286		

11.4 用结构体类型解决问题	347	13.2.2 头文件设计的忠告	399
实例研究: 用于复数的用户自定义类型	347	13.3 个人库: 实现文件	400
11.5 并行数组和结构体的数组	352	13.4 存储类	402
11.5.1 并行数组	352	13.4.1 全局变量	403
11.5.2 结构体数组的声明	352	13.4.2 static和register存储类	404
实例研究: 通用计量转换	353	13.5 库中包含函数的修改	405
11.6 共用体类型(可选)	359	13.6 条件编译	407
11.7 常见编程错误	363	13.7 函数main的参数	410
复习	364	13.8 定义带参数的宏	411
快速测试	365	13.8.1 在宏体中使用括号	413
快速测试答案	365	13.8.2 在两行以上扩展宏	414
复习题	366	13.9 常见编程错误	415
程序设计项目	367	复习	415
第 12 章 文本文件和二进制文件的		快速测试	417
处理	372	快速测试答案	417
12.1 输入/输出文件: 回顾与进一步研究	372	复习题	418
12.1.1 键盘和屏幕作为文本流	372	程序设计项目	418
12.1.2 换行与EOF	373	第 14 章 动态数据结构	421
12.1.3 转义序列	373	14.1 指针	421
12.1.4 printf的格式输出	373	14.1.1 作为函数参数的指针	422
12.1.5 文件指针变量	374	14.1.2 表示数组和字符串的指针	422
12.1.6 获取文件指针参数的函数	375	14.1.3 指向结构体的指针	423
12.1.7 关闭文件	376	14.1.4 指针使用的小结	423
12.2 二进制文件	378	14.2 动态内存分配	424
12.3 数据库查询	382	14.2.1 访问动态分配的结构体成员	425
实例研究: 数据库查询	382	14.2.2 使用calloc动态分配数组	426
12.4 常见编程错误	389	14.2.3 返回堆单元	426
复习	390	14.3 链表	428
快速测试	391	14.3.1 带指针成员的结构体	428
快速测试答案	391	14.3.2 连接结点	429
复习题	392	14.3.3 链表的优点	431
程序设计项目	392	14.4 链表运算符	432
第 13 章 大型程序设计	395	14.4.1 遍历链表	432
13.1 使用抽象处理复杂问题	395	14.4.2 获得输入表	434
13.1.1 过程抽象	395	14.4.3 在表中查询目标	435
13.1.2 数据抽象	396	14.4.4 避免跟踪NULL指针	435
13.1.3 信息隐藏	396	14.5 用链表表示栈	436
13.1.4 可重用代码	396	14.6 用链表表示队列	438
13.2 个人库: 头文件	397	14.7 有序表	443
13.2.1 头文件	398	实例研究: 维护一个整数有序表	443
		14.8 二叉树	451

14.8.1 二叉查找树	452	复习	493
14.8.2 搜索二叉查找树	452	C语言结构的复习	493
14.8.3 建立二叉查找树	454	快速测试	494
14.8.4 显示二叉查找树	456	快速测试答案	495
14.9 常见编程错误	458	复习题	495
复习	459	程序设计项目	496
快速测试	460		
快速测试答案	461	第 16 章 关于 C++	497
复习题	461	16.1 C++控制结构、输入/输出以及函数	497
程序设计项目	462	16.1.1 使用名字空间std	497
第 15 章 使用进程和线程的多进程	464	16.1.2 C++标准输入/输出	500
15.1 多任务	464	16.1.3 引用参数	500
15.1.1 串行程序设计和并行程序 设计	464	16.1.4 输出格式化	502
15.1.2 分时多任务	465	16.2 C++对面向对象编程的支持	503
15.1.3 抢占式多任务	465	16.2.1 头文件complex.h	504
15.1.4 时间片和并行	466	16.2.2 类名和构造函数	505
15.1.5 并发程序设计	467	16.2.3 成员函数和运算符	506
15.2 进程	467	16.2.4 实现文件complex.cpp	506
15.2.1 进程创建	467	16.2.5 数据成员	509
15.2.2 进程等待	469	16.2.6 输入/输出运算符重载	510
15.2.3 从进程中执行另一个程序	470	复习	512
15.3 进程间通信和管道	471	快速测试	513
15.3.1 管道	471	快速测试答案	514
15.3.2 管道的用法	472	复习题	514
15.3.3 使用标准输入的进程间通信	473	程序设计项目	515
15.3.4 父进程和子进程间通信举例	474	附录 A 字符集	516
15.4 线程	476	附录 B ANSI C 标准库	518
15.4.1 线程的创建	477	附录 C C 运算符	529
15.4.2 线程的同步	478	附录 D 关于指针的更多信息	533
15.4.3 互斥锁	479	附录 E ANSI C 保留字	537
15.4.4 死锁	481	附录 F ANSI C 结构的参照指南	538
15.5 线程举例	484	词汇表 (图灵网站下载)	
实例研究: 生产者/消费者模型	484	索引 (图灵网站下载)	
15.6 常见编程错误	493	自测练习奇数题答案 (图灵网站下载)	

计算机与程序设计概述

21世纪的人类生活由无数计算机主导。从早上自动煮咖啡的咖啡壶，到烹调早餐的微波炉，到上班路上所驾驶的汽车，再到提取现金的自动提款机，人类生活的各个方面都要依赖计算机。计算机接收、存储、处理并输出信息，它可以处理的数据包括数值、文本、图像、图形和声音等各种形式。

计算机 (computer) 能够接收、存储、转换并输出各种数据的机器。

在计算机技术中，计算机程序的作用非常重要。如果没有一系列的指令，那么计算机实际上毫无用处。利用程序设计语言可以编写程序，并与计算机交流。

本书将利用目前最通用的程序设计语言之一——C语言来介绍计算机科学。本章介绍计算机及其组成，以及程序设计语言的主要分类。本章还论述了计算机如何处理C语言，并且描述了一种用来解决编程问题的称为软件开发方法的系统方法及其应用。

1.1 电子计算机的过去和现在

在日常生活中，我们会经常接触到计算机。有些人利用计算机来做字处理，有些人甚至在中学已经学习了编程，但以前并不是这样。在不久以前，大多数人还认为计算机是一种神秘设备，其秘密只被少数计算机奇才掌握。

第一台电子计算机是由John Atanasoff博士和Clifford Berry于20世纪30年代后期在艾奥瓦州立大学创建的。Atanasoff设计该计算机是为了帮助核物理专业的研究生完成数学计算。

第一台大规模的通用数字电子计算机（称之为ENIAC）由美国军方资助，于1946年在宾夕法尼亚大学问世。该计算机重30 t，占地面积为139m²。ENIAC用于计算弹道表、预测天气并进行原子能计算。

这些早期的计算机使用电子管作为基本的电子元件。电子元件的设计和制造技术的发展使得新一代计算机比以前的计算机更小、更快、更便宜。

利用如今的技术，计算机处理器的整个电路可以封装成一个邮票大小的电子元件，称之为计算机芯片或微处理器芯片（图1-1）。计算机芯片价格便宜，尺寸很小，可以安装在手表、便携式计算机、照相机、家用电器、汽车当然还有计算机上。

2

计算机芯片 (微处理器芯片) (computer or microprocessor chip) 包含计算机处理器集成电路的硅片。

今天，个人计算机已走进办公室和寻常百姓家，通常价格不会超过1 000美元，并且可以放在桌上。这跟20年前花费超过100 000美元，并且占据10m²空间的计算机所具备的计算能力是一样的。更小的计算机还可以放在公文包内（图1-2a）甚至手中（图1-2b）。

现代计算机根据其大小和性能分类。个人计算机（如图1-2所示）一次只允许一人使用。巨大的实时交易处理系统（例如ATM和其他银行网络）以及汽车旅馆、航空公司和汽车租赁公司用的预订系统则需要使用强有力的、可靠的大型机。最大容量和最快的大型机称为超级计算机，它用于研究型实验室和计算量密集的场所，如天气预报。

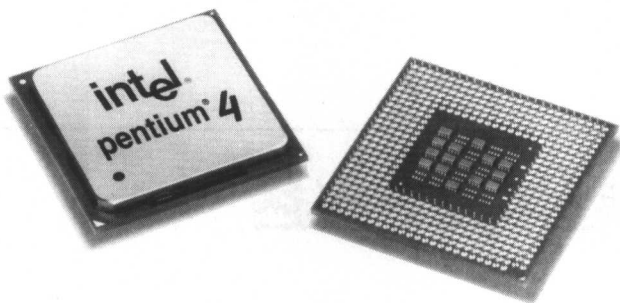


图1-1 Intel Pentium 4处理器芯片是一个包含中央处理单元完整电路系统的集成电路。这种处理器可以在60亿分之一秒内执行一条类似整数加法这样的简单指令（经Intel公司允许翻印，©Intel公司，2003）

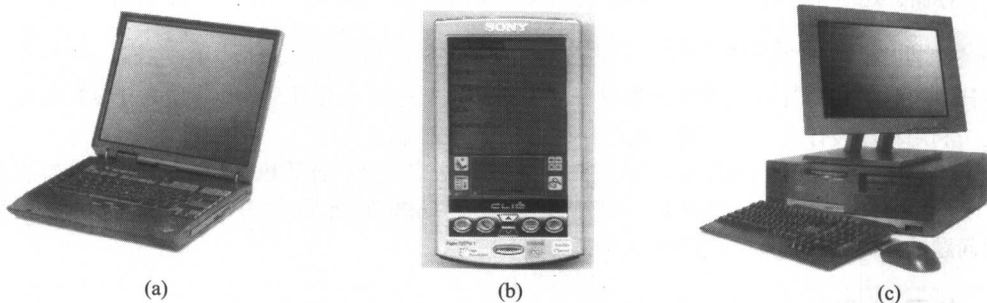


图1-2 (a)笔记本电脑（ThinkPad[®]，经IBM公司允许），(b)掌上型计算机（Sony Clie PDA[®]，经Sony公司允许），(c)台式计算机（IBM NetVista Desktop，经IBM公司允许）

3
~
4

构成计算机系统的要素主要分为两大类：硬件和软件。硬件是用于完成必要计算的设备，它包括中央处理器（CPU）、显示器、键盘、鼠标、打印机和扬声器。软件由程序组成，程序通过给计算机提供一系列要执行的指令来解决问题。

硬件（hardware） 实际的计算机设备。

软件（software） 与计算机相关的程序集合。

程序（program） 使计算机执行具体任务的一系列指令。

计算机编程在这些年已经经历了巨大的变化。最初编程非常困难，要求程序员以很长的二进制数（01序列）编写程序指令。C语言这样的高级程序设计语言的出现，使得程序设计变得容易多了。

二进制数（binary number） 由0和1数字序列构成，以2为基的数。

练习

自测

1. 计算机程序是硬件还是软件？
2. 大型机用于什么场合？

1.2 计算机硬件

尽管在价格、大小和性能方面有很大不同，但现代计算机在许多基本方面是相似的。实质上，大多数计算机都包含下面的组成部件：

- 主存储器（主存）。
- 辅助存储器（外存），包括硬盘、软盘、压缩盘、CD和DVD等存储设备。