计 算 机 科 学 丛 书

C程序设计教程

C how to program

Second Edition

(美) H. M. Deitel
P. J. Deitel 著
薛万鹏 等译

机械工业出版社
China Machine Press
Prentice Hall

计算机科学丛书

C 程序设计教程

(美) H.M.Deitel
P.J.Deitel 著
薛万鹏 等译
张祖荫 审校

机械工业出版社
China Machine Press

本书详细叙述了C程序设计语言，强调用结构化程序设计方法编写程序，自始至终用完整的程序输出范例来演示所讲的概念，内容全面，层次清晰，可作为大专院校学生和计算机编程爱好者的入门与提高教程。

本书版权登记号：图字：01－97－0506

图书在版编目（CIP）数据

C程序设计教程/（美）迪特尔（Deitel, H.M.），（美）迪特尔（Deitel, P.J.）著；薛万鹏等译．－北京：机械工业出版社，2000.7
（计算机科学丛书）
书名原文：C How To Program, Second Edition
ISBN 7－111－07952－3

Ⅰ.C…　Ⅱ.①…②迪…③薛…　Ⅲ.C语言－程序设计－教材　Ⅳ.TP312

中国版本图书馆CIP数据核字（2000）第16458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：赵红燕
北京昌平第二印刷厂印刷·新华书店北京发行所发行
2000年7月第1版第1次印刷
787mm×1092 mm　1/16·31.75印张
印数：0 001-6 000册
定价：33.00元

出 版 说 明

《C/C++程序设计大全》是国外最受欢迎的大学教科书之一，采用率很高，已由机械工业出版社于1997年8月翻译出版，读者反映良好。为满足大专院校师生学习要求和社会上广大读者的需要，现在我社将此书重新修订，分为两册出版，书名为《C程序设计教程》和《C++程序设计教程》。

译 者 序

C语言表达能力强，目标代码效率高，可移植性好，既具有高级语言的优点，又具有低级语言的特点，因此特别适合编写操作系统、网络软件、编译器等系统软件。

市面上介绍C语言的书是很多的，这些书的内容可以说是“大而全”。C语言入门并不难，但是，正如许多人认为的那样，掌握程序设计语言最困难之处是用其灵活高效地开发实际软件系统，这需要大量的实践和学习，而C因其丰富的功能和复杂的特点更是如此。

与其它书不同，本书从软件工程的角度介绍并讨论了C语言，读者能在学习的同时为实际使用打下坚实的基础，初学者和有经验的程序员都会从中受到启发。原书作者有着丰富的软件开发经验。Harvey M.Deitel教授是虚拟存储系统（VMS）的先驱研究者之一，如今这种系统已经广泛地应用于UNIX、OS/2和Windows NT等等的操作系统上；Paul J.Deitel在网络数据管理、数据库查询翻译器、金融财务管理系统的开发方面具有丰富的经验。本书的许多内容都体现了作者对程序设计技术的理解，正如作者所言：“本书集我们40余年程序设计经验之精华”。

本书主要由薛万鹏、纪宁、韩磊、许文轩、梅开、谢立、薛鸾、李岩、沈长华翻译，参加翻译工作的还有濮玉民、赵晓蓉、颜先杰、任映梅、治中、瞿跃龙、苏泳民、史荣光、上官立新、单力、汪梓鸣、成文、魏莲方、马蔚、杨开开、段群慧、蒋星、文达、韩青云等，张祖荫、梁敏、沈维亮、李昕怡对本书进行了全面的审校。机械工业出版社华章公司和华译工作室也为本书的翻译给予了大力的支持和帮助，在此深表感谢。

原书内容严谨，具有较强的理论性和实用性。译者力求反映原书的特点和风貌，但由于时间关系及水平所限，不当和疏漏之处在所难免，敬请广大读者批评指正。

1997年4月

前　言

欢迎进入C语言世界！本书是由父子两代人合著而成。年长者H.M.Deitel编写、教授程序设计已30余年，经验丰富、强调程序的清晰优美。年轻人P.J.Deitel编写程序亦有十余年，精力充沛、热衷于教学和程序设计，侧重于程序的性能和结果。我们希望通过我们两人的合作，奉献给读者一本集知识性和可读性于一体的书。

C语言通常只教授给有程序设计基础的人。许多教育家认为C语言的复杂性及其众多的难点使得该语言不适合作为程序设计的第一门教学语言，而这正是本书的目标。那么我们为什么还要编写这本书呢？

事实上，C语言已成为工业界选用的系统实现语言。在大学教授Pascal语言已13年的H.M.Deitel强调清晰的结构化程序。Pascal课程中所讲的许多内容是结构化程序的基本原则。我们以H.M.Deitel在大学里的教学方式编写了这本教材。我们会指出疑点并说明有效解决的步骤。我们的经验是：要以学习Pascal语言同样的方式学习该教材。不过有一个值得注意的不同点，那就是学生们正为他们在学习一种毕业后就能立即有用的语言所鼓舞，这提高了他们的学习热情，在想到C语言难学时，这种学习热情对克服困难大有帮助。

出版本书的目的是为那些没有或很少程序设计经验的学生提供相当于大学水准的入门教程，同时也是要对传统C教程所要求的理论和实践作严密的论述。为了达到这些目标，本书的内容比其它C语言教材丰富得多。在我们的教学中约有1000名学生使用了该教材，好几万名外国读者详细地学习了本书的第一版。

为了让读者能够解决生动有趣的现实问题，本书提供丰富的范例、练习和从许多领域中提炼出的专题。

本书侧重于软件工程的原则，强调用结构化程序设计方法编写清晰的程序，力求避免使用模糊的术语和语法规范。

本书用完整的程序和输出范例来演示所讲的概念。每一章以学习目标开头，最后以小结和本章涉及的术语结束。每一章还配有习题。本书的练习涵盖面广，既有简单的复习题，也有复杂的程序设计问题，还有大型项目的开发。为了提高本书对学生的价值，我们在练习中投入了大量的精力。本书中的程序已在ANSI C兼容的编译器上作了测试，所用的机器包括Sun SPARC工作站、Apple Macintosh（Think C）IBM PC（Turbo C++，BorlandC++）及DEC VAX/VMS（VAX C）。

关于本书

这本书具有帮助学习的如下特点：

学习目标

每一章均以学习目标开头，它告诉学生应该达到什么标准，并能够在学完该章后测试自己是否达到了标准。

节

每章均由重点明确的节组成。每一节在介绍完整的C程序的同时介绍了C语言的特征。每一个程序之后都有输出结果，输出结果证实了所介绍的内容。把输出结果和程序设计语句结合起来讲解是学习和巩固所讲概念的好方法。本书的程序是为反映C语言的种种特点而设计的，在认真阅读的同时还需要上机实践。

图解

本书具有丰富的图解。结构化流程图有助于理解控制结构和结构化程序设计。本书还用大量的图示来说明重要数据结构（如链表、队列、堆栈、二叉树）的建立和维护。

小结

每章的小结部分的内容可帮助学生复习和巩固重要的概念。

术语

每一章中还按字母顺序列出了本章定义的术语，它也是为了让学生进一步巩固所学的概念。然后归纳了良好的程序设计习惯、常见的程序设计错误、性能忠告、可移植性忠告及软件工程评述。

自我测验练习

为便于自学，本书还配有自我测验练习及答案。它可帮助学生树立信心并尝试其后的练习。

练习

每一章配有大量的练习。这些练习涵盖面广，包括重要术语及概念的回忆、编写一条C语句、写一小段C函数、编写完整的C函数和程序、直至编写重大项目。教师可根据需要和各学期的教学计划加以选择，还可以用这些练习题来安排家庭作业、小测验及重要考试。

本书概况

本书共有14章和5个附录，详细地叙述了C程序设计语言（包括结构化程序设计的内容）并提供了支持正文的参考资料。

第1章“基本概念”，讨论什么是计算机、它是如何工作的以及人们是怎样为计算机编写程序的。引入了结构化程序设计的概念，解释了为什么这种技术促成了程序设计方法的革命。叙述了从机器语言到汇编语言及高级语言的发展史。介绍了C语言的起源和C程序设计的环境。

第2章“C语言程序设计入门”，简要地介绍了C程序的编写，详述了C语言中的判断及算术运算。学完这一章后，学生将理解怎样编写简单而完整的C程序。

第3章“结构化程序的开发”，它可能是本书最重要的一章。本章介绍了帮助解决问题的算法表示，解释了结构化程序设计在编写易于理解、调试、维护及更可能在第一次测试就能正确运行的程序中的重要性。介绍了结构化程序设计的基本控制结构，即顺序结构、选择结构和循环结构。解释了自顶向下、逐步求精的程序设计技术，这种技术对编写正确的结构化程序是很重要的。介绍流行的帮助设计程序的方法——结构化伪码。第3章介绍的方法不仅适用于C语言，而且也适用于其它任何结构化程序设计语言。本章有助于培养学生良好的程序设计习惯，从而为解决复杂程序设计任务打下坚实的基础。

第4章“程序控制”，提炼了结构化程序设计的表示方法并介绍了其它控制结构。详细

地探讨了循环结构，对比了用计数器控制的循环和用标记控制的循环。介绍了 for 结构，它是实现用计数器控制的循环的一种方便的方法。还介绍了 switch 选择结构和 do/while 循环结构。最后还讨论了逻辑运算符。

第 5 章“函数”，讨论了程序模块的设计和构造，介绍的内容包括标准库函数、程序员定义的函数、递归和传值调用。本章介绍的技术对于开发正确的结构化程序（尤其是大型程序和软件）相当重要。本章所介绍的“细化”程序设计策略是解决复杂问题的有效的方法，这种策略就是用函数把复杂的问题分解为简单而又相关的部分。学生们欣赏本书对随机数和模拟的介绍，还欣赏本书对完美地使用了控制结构的掷骰子游戏的讨论。本章还介绍了递归结构，列出了书（包括练习）中涉及到的 31 例递归范例。有些教材把递归结构放在后面介绍，但我们认为这一内容最好逐步覆盖全书。练习中搜集了几个经典的递归问题，如汉诺塔问题（Towers of Hanoi）。

第 6 章“数组”，讨论了数组数据结构（即同一类型的相关数据项的集合）。本章列举了大量的一维数组和二维数组的范例，阐明了结构化数据与控制结构对于正确地开发结构化程序同样重要。本章的范例涉及到对数组的各种常用操作方法、打印直方图、数据排序、把数组传递给函数以及统计数据的简单分析。本章的特点之一是仔细地介绍二分查找法，这种方法是对线性查找方法的极大的改进。练习中有许多有趣而富有挑战性的问题，包括排序方法的改进、航空预约系统的设计、龟图概念（在 LOGO 语言中很著名）和在人工智能领域广泛作为程序开发概念的骑士漫游和八皇后问题。

第 7 章“指针”，讲述了 C 语言中功能最强大的特点之一——指针。详细地解释了指针的操作、传引用调用、指针表达式、指针的算术运算、指针与数组的关系、指针数组和指向函数的指针。练习中有经典的龟兔赛跑的模拟、洗牌和发牌的算法，还包括专题“建立自己的计算机”。专题中解释了机器语言程序设计的表示方法，让学生设计和实现一个允许编写和运行机器语言程序的计算机模拟器。这种特点对于那些希望理解计算机到底是如何工作的读者尤其有用。学生们喜欢这种专题，他们经常能够补充一些增强功能（练习中建议了许多增强功能）。在第 12 章中，另一个专题指导了读者建立一个编译器，然后把编译器所生成的机器语言放在第 7 章中建立的机器语言模拟器上运行。

第 8 章“字符和字符串”，探讨了处理非数值数据的基本问题。这一章完整地介绍了 C 函数库中的字符和字符串处理函数。所讨论的技术可广泛地用于建立字处理器、页版面和排版软件、文本处理应用程序。本章搜集了 33 道与文本处理有关的有趣的练习题，涉及的领域包括回文、打油诗、英文转换、产生表示给定电话号码的七字单词、文本对齐、用单词书写支票金额、产生莫尔斯码、公制转换和讨债信。本章最后一道练习是对学生的挑战，它要求用计算机化的字典建立纵横字谜发生器。

第 9 章“格式化输入/输出”，介绍了 printf 和 scanf 函数的强大的格式化能力。讨论了 printf 函数的格式化输出功能，如把浮点数精确到指定的小数位、数值的列对齐、左对齐和右对齐、插入直接信息、强制打印出加号、打印前导零、用指数表示法输出结果、八进制和十六进制的使用、以及域宽和精度的控制。讨论了 printf 函数的转义序列，如光标的移动、特殊字符的打印和系统响铃等等。讨论了 scanf 函数的所有的格式化输入能力，包括读取十进制、八进制、十六进制、浮点数、字符和字符串数据。还讨论了通过扫描输入的数据从扫描集中读取匹配或不匹配的字符。本章的练习实际上测试了 C 的所有的格式化输入/输出能

力。

第 10 章“结构、联合、位运算和枚举”，介绍了各种重要的特点。结构类似于 Pascal 语言和其它语言中的记录，它把各种类型的数据项组合在一起。第 11 章用结构来建立包含以记录格式组织信息的文件。第 12 章把结构和指针以及动态内存分配一起使用，建立了链表、队列、堆栈和树等等的动态数据结构。联合用于使一块内存区能够被不同类型的数据在不同时刻使用。枚举对定义有用的符号常量是很方便的，它能够使程序具有更好的可读性。C 语言的功能强大的位操作能力能够让程序员编写控制底层硬件的程序。程序中使用位操作能够处理位串、屏蔽（或取消屏蔽）某个位和更紧凑地存储信息。这种通常只在底层汇编语言中才有的能力可以让程序员编写系统软件（如操作系统和网络软件）。本章的特点之一是重写了以前的洗牌和发牌模拟程序，新的程序效率更高，从中可以看出算法质量的重要性。

第 11 章“文件处理”，讨论了处理顺序存取文本文件和随机存取文本文件的技术。先介绍了数据的层次结构，即从位、字节、字段、记录到文件这样的一种层次结构。然后简要地介绍了 C 语言把文件看作字节流的观点。本章用三个程序讨论了顺序存取文件，这些程序说明了怎样打开和关闭文件、怎样按顺序把数据存储在文件中以及怎样按顺序读取文件中的数据。讨论随机存取文件时介绍了四个程序，这些程序说明了怎样按顺序建立一个随机存取的文件、怎样随机地读取和写入文件数据、怎样按顺序读取随机存取文件中的数据。第四个程序是一个完整的金融事务处理程序，它综合了顺序和随机存取文件的许多技术。工业界的读者告诉我们说，通过本书的学习，他们已经能够编写在其机构内适用的文件处理程序。

第 12 章“数据结构”，讨论了用于建立动态数据结构的技术。本章以讨论自引用结构和动态内存分配开始，然后讨论了怎样建立和维护各种动态数据结构，包括链表、队列、堆栈和树。对于每一种类型的数据结构，我们都提供了完整的程序和输出结果。第 12 章有助于学生真正地掌握指针。本章包含了使用间接引用和二次间接引用（特别难懂的概念）的丰富的范例。指针学习的难点之一是学生难以想象数据结构以及结点的连接。为此，我们用图示的方法来说明链节及其建立的顺序。用二叉树范例研究指针和动态数据结构是再好不过了。范例程序建立了一个二叉树，并以先根次序、中根次序和后根次序遍历该二叉树。学生们特别欣赏以中根次序遍历并打印结点的排序值。本章有丰富的练习，其中的重点之一是专题内容“建立自己的编译器”。该专题练习指导学生开发把中缀表达式转换为后缀表达式的程序和计算后缀表达式的程序。然后，我们修改了后缀表达式的算法，并用该算法产生机器语言代码。编译器把它所产生的机器语言代码文件放在了一个文件中（用第 11 章中的方法），学生可以在第 7 章开发的软件模拟器上运行编译器所产生的机器语言代码。

第 13 章“预处理程序”，详细地讨论了预处理程序的指令。本章完整地介绍了预处理指令 #include 和 #define。#include 指令在程序编译之前把指定文件的拷贝包含到指令所在的位置，#define 指令用来建立符号常量和宏。本章解释了条件编译，它能够让程序员控制预处理指令的执行和对源代码的编译。还讨论了把其操作数转换为字符串的运算符 # 和连接两个记号的运算符 # #，介绍了五个预定义的宏常量 __LINE__、__FILE__、__DATE__、__TIME__和__STDC。最后讨论了头文件 assert.h 中的宏 assert，在测试、调试和校验程序时，宏 assert 是有价值的。

第 14 章“高级话题”，介绍了在通常的教程中不涉及到的某些高级话题。第 14.2 节介绍了怎样把来自于文件的输入重定向到程序中、怎样把来自程序的输出重定向到文件中、怎

样把一个程序的输出重定向成另一个程序的输入（即建立管道）以及怎样把程序的输出追加到现有的文件中。第 14.3 节讨论了怎样开发使用变长参数列表的函数。第 14.4 节介绍了怎样把命令行参数传递给函数 main 以及怎样在程序中使用这些参数。第 14.5 节讨论了怎样编译由多个源文件组成的程序。第 14.6 节讨论了用函数 atexit 注册要在程序结束时执行的函数，还讨论了结束程序执行的函数 exit。第 14.7 节讨论了类型限定符 const 和 volatile。第 14.8 节介绍了怎样用整数和浮点数后缀指定数值常量的类型。第 14.9 节解释了二进制文件和临时文件的用法。第 14.10 节介绍了怎样用信号处理库捕捉意外的事件。第 14.11 节讨论了用函数 calloc 和 realloc 建立和使用动态数组。

附录部分提供了有价值的参考资料。附录 A 是 C 语法概要，附录 B 摘编了 C 标准库，附录 C 完整地列出了运算符的优先级和结合性，附录 D 列出了 ASCII 字符集代码，附录 E 讨论了二进制、八进制、十进制和十六进制数值系统。附录 B 是根据 ANSI 标准文献摘编的（经美国国家标准化协会允许），该附录对编写 C 程序是详细而有价值的。附录 E 是一套完整的数值系统，其中包括自我测验练习和答案。

欢迎使用本书。本书作者期待您的意见、批评、指正和建议。

目　录

第1章 基本概念

学习目标：

- 理解计算机的基本概念
- 熟悉程序设计语言的不同类型
- 熟悉C语言的历史
- 知道C标准库
- 理解C程序的开发环境
- 认识把C作为第一门程序设计语言课程为什么是合适的
- 认识C语言为什么是进一步学习程序设计特别是C++的基础

1.1 引言

欢迎进入C语言世界！我们真诚地希望本书能够给你提供内容丰富、形式活泼的C语言知识。C是一门难以掌握的语言，通常只有富有经验的程序员才去学习它。与其它C语言教材相比，本书具有如下特色：

- 适合很少或没有程序设计经验的读者。
- 适合有一些经验但还想系统深入地学习C语言的程序员。

为了吸引这两类读者，本书把重点放在这两类读者共同感兴趣的方面，即怎样用结构化程序设计技术编写清晰的程序。通过本书的学习，没有程序设计经验的读者能够从一开始就以正确的方法设计程序。为了帮助读者准确地理解所讲的内容，本书配有丰富的插图，并提供了大量的范例程序和运行结果。

前四章介绍了计算机、程序设计和C语言程序设计的基础知识，穿插了结构化程序设计的内容。初学者告诉我们说，这几章为深入地学习第5章到第14章的内容打下了坚实的基础。有经验的程序员通常可快速地阅读前四章的内容，阅读完这些内容后，他们会发现第5章到第14章的内容是严谨而富有挑战性的。他们特别欣赏后几章对指针、字符串、文件和数据结构的细节的介绍。

许多有经验的程序员告诉我们说，他们欣赏我们对结构化程序设计内容的介绍。虽然他们经常用Pascal等等的结构化语言设计程序，但是因为他们从来没有正式学习过结构化程序设计，所以不能编写出最好的代码。学完本书后，他们已经改用了良好的程序设计风格。因此，不论你是新手还是有经验的程序员，你都能从本书学到许多新的内容，并且还会面临许多新的挑战。

大多数人都知道计算机能完成许多激动人心的工作，本书要向你介绍怎样命令计算机完成这些任务。控制计算机的是软件，也就是你所编写的命令计算机完成某些动作和决策的指令，而C语言是当今最流行的软件开发语言之一。本书介绍ANSI C程序设计技术。ANSI C是1989年推出的标准化C版本，它既是美国国家标准化协会（ANSI）公布的美国标准，

也是国际标准化组织（ISO）公布的世界标准。

计算机在各个领域的应用正在迅猛发展。在当今成本不断上升的时代，计算机的成本却因为硬件和软件技术的迅猛发展而大幅度地下降。25年前需要一个大房间才能容纳得下、成本在几百万美元的计算机如今能够做在一个比手指甲还小的硅片上，其成本可能只有几美元。有意思的是，硅是地球上最丰富的材料之一。硅片技术的发展使计算机成本如此低廉，从而使全球大约有1.5亿台通用计算机应用于商业、工业、政府机构和为个人所用。这个数字可能会在未来几年内成倍上升。

能不能把C语言作为第一门程序设计的教学语言呢？我们认为是可以的。两年前，当Pascal语言牢居第一门教学语言位置的时候，我们编写了本书的第一版。基于本书的教学证实，把C语言作为第一门教学语言同样有效。这两门课程的教学效果没有明显的差别，而C语言的教学更能激发学生的学习热情，因为他们知道高年级的课程以及他们以后的职业生涯中更可能用C语言而不是Pascal。学习了C语言的学生还认识到，学好C语言能够使他们更快地掌握C++。C++用来编写面向对象的程序，它是C的超集。

好，让我们开始学习吧！你将开始面临一场新的挑战并且会得到充分的回报。

1.2　计算机是什么

计算机是能够以比人快几百万甚至几十亿倍的速度执行计算和逻辑判断的设备。例如，当今许多个人计算机能够在一秒钟内执行几千万次的加法运算。使用桌面计算机的人可能需要用几十年的时间才能完成功能强大的个人计算机在一秒钟内所完成的计算量（值得思考的问题：怎样知道这个人是否正确地完成了加法计算？又怎样知道计算机是否正确地完成加法运算?）。当今最快的超级计算机能够在每秒钟内执行几千亿次的加法计算，大约相当于几十万人一年的计算量！每秒执行几十亿次指令的计算机已经应用于科研机构。

计算机在称为“计算机程序”的指令集控制下处理数据。计算机程序控制着计算机，使它按顺序执行一系列动作，这些动作是由称为“计算机程序员”的人员指定的。

构成计算机系统的各种设备称为“硬件”，如键盘、屏幕、磁盘和处理单元等等。在计算机上运行的程序称为“软件”。硬件成本在最近几年大幅度地下降，个人计算机已经成为日用品。不幸的是，由于软件开发技术没有像硬件那样得到相应的提高，软件成本随着程序员开发功能更强大、系统更复杂的应用程序而稳步上升。本书要介绍降低软件开发成本和缩短功能强大的高质量软件开发过程的软件开发技术。这些方法包括结构化程序设计、自顶向下逐步求精的开发方法、功能化设计等等。

1.3　计算机的结构

各种计算机不论外观差别有多大，每一台计算机实际上都可划分为六个逻辑单元（或称为六大部分）。

1）输入单元　输入单元是计算机中接收信息的部分。它从各种输入设备读取信息（数据和计算机程序），并把这些信息放置到其它处理信息的单元中。当今计算机的大多数信息是通过键盘输入的。

2）输出单元　输出单元是计算机中输出信息的部分。它把计算机处理过的信息放置到各种输出设备中，从而使这些信息能够被计算机外部使用。当今计算机的大多数信息是通过

屏幕显示和纸张打印输出的。

3）内存单元 内存单元是计算机中存取速度快、容量相对较低的存储部分。它能记忆来自输入单元的信息，因而能够在需要的时候立即处理这些信息。内存单元还能记忆被处理过的信息，直到输出单元把信息放到输出设备上。内存单元经常被称为“内存”或“主存”。

4）算术逻辑单元（ALU） 算术逻辑单元是计算机中的“加工”部分。它执行加法、减法、乘法和除法等等的计算。ALU 中包含判断结构，如比较内存中的两项是否相等。

5）中央处理单元（CPU） 中央处理单元是计算机中的“管理”部分。它是计算机的协调者，负责管理对其它部分的操作。在应该把信息读到内存单元中时，CPU 就会给输入设备发出请求；当要把内存中的信息用于计算时，CPU 就会把这种请求通知给 ALU；当要把内存中的信息发送给某个输出设备时，CPU 就会告诉输出设备。

6）二级存储单元 二级存储单元是计算机中长期保存信息的高容量存储部分。没有被其它单元使用的程序和数据通常放在二级存储设备中（如磁盘），这些信息可能需要几个小时、几天、几个月甚至几年才会被使用。

1.4 批处理、多道程序设计和分时

早期的计算机在同一时间只能执行一个作业或任务，这种运作方式通常称为单用户批处理方式。计算机在处理成组或成批的数据时，在同一时间只能运行一个程序。在早期的批处理系统中，用户通常用穿孔卡片组把作业提交到计算机中心，并且要等待数小时或几天才能获得输出结果。

随着计算机的功能越来越强大，单用户批处理方式利用计算机资源的效率越来越低。于是人们想到了用多个作业或任务分享计算机资源。这种处理方式称为“多道程序设计”。多道程序设计可在计算机上“同时”运算多个作业，计算机在这些作业之间分享其资源。对于早期的多道程序设计系统，用户仍然要用穿孔卡片组提交作业并且要等待几个小时或几天才能获得输出结果。

在 60 年代，几个来自工业界和大学的研究小组率先研究了“分时”的概念。“分时”是多道程序设计的特例，利用分时技术的用户通过输入/输出设备或终端访问计算机。典型的分时计算机系统可能会有几十个甚至上百个用户同时共享它。计算机实际上并不是同时为所有用户服务，而是在运行完某个用户的一小部分作业后，再为下一个用户服务。计算机运行得太快了，它能够在一秒钟内为每个用户服务多次，因此用户感觉不到计算机在同时为多个用户服务。

1.5 个人计算、分布式计算和客户/服务器计算

1977 年，苹果计算机公司（Apple Computer）开始推出个人计算机，这是计算机业余爱好者最初的梦想。随着计算机越来越便宜，人们已经有能力为个人使用或商业使用而购买它了。1981 年，世界上最大的计算机供应商 IBM 公司推出了 IBM 个人计算机。昨天的梦想成为今天的现实，个人计算机正式走进了商业、工业和政府机构。

但是，这些计算机是独立的设备，所以人们先在自己的计算机上完成自己的工作，然后通过磁盘共享信息。尽管早期的个人计算机没有足够强大的功能在几个用户之间分时共享资源，但是这些计算机可以通过计算机网络连接在一起（有时是通过电话线连在一起的，有时

是连在某个机构的局域网上)。这就产生了分布式计算，即某个机构的计算量通过网络分给完成实际工作的计算机去完成，而不是局限于让某个中央计算机去完成。个人计算机能够处理单用户的计算需求和基本的通信任务。

当今功能最强大的个人计算机，其功能已经相当于10年前的几百万美元的计算机。功能最强大的桌面计算机（称为“工作站”）能够给单用户提供大量的功能。信息能够很容易地通过计算机网络共享。网络中的某些称为“文件服务器”的计算机存储了供分布在网络上的客户计算机使用的公用的程序和数据，这就产生了“客户/服务器计算”。C已经成为编写操作系统软件、计算机网络软件和分布式客户/服务器应用程序的程序设计语言。

1.6　机器语言、汇编语言和高级语言

程序员用各种程序设计语言编写计算机指令。某些指令能够直接被计算机执行，而其它的指令还需要通过中间的翻译过程。当今使用的计算机语言有上百种，大致可分为如下三类：

1）机器语言。

2）汇编语言。

3）高级语言。

所有的计算机都只能直接执行其自身的机器语言。机器语言是特定计算机的“自然语言”，它与计算机硬件的设计密切相关。机器语言通常是由指示计算机一次完成一个最基本操作的数值串组成的（这些数值最终要转换为若干个1和0)。机器语言是与机器有关的，即特定的机器语言只能用在特定的一类计算机上。下面的机器语言程序把加班工资和基本工资相加，结果存储在总工资中。从这个程序段可看出，机器语言是繁琐的。

```
+1300042774
+1400593419
+1200274027
```

随着计算机越来越受到人们欢迎，对大多数程序员来说，用机器语言设计程序显然太慢并且非常枯燥。为了取代计算机能够直接理解的数值串，程序员开始使用代表计算机基本操作的类英语缩写来编写程序。这些类英语缩写构成了汇编语言的基础。为了把汇编源程序转换为机器语言，人们开发出了称为“汇编程序”的“翻译程序”。下面的汇编源程序也把加班工资和基本工资相加并把结束存储在总工资中，但它比相应的机器语言清晰得多。

```
LOAD    BASEPAY
ADD     OVERPAY
STORE   GROSSPAY
```

随着汇编语言的出现，计算机的用途迅速扩大，但是即使完成最简单的任务还仍然需要编写许多指令。为加速程序开发的进程，人们开发出了一条语句能够完成一定意义的任务的高级语言。把高级语言程序转换为机器语言的翻译程序称为“编译器”。用高级语言编写的指令非常类似于日常英语并且包含常用的数字符号。用高级语言编写的工资表程序可能会含有类似于如下的语句：

```
grossPay = basePay + overtimePay
```

显然，从程序员的角度看，高级语言比机器语言和汇编语言更令他们满意。C是功能最强大、使用范围最广的高级语言之一。