



普通高等教育“十一五”国家级规划教材
普通高等教育计算机基础课程规划教材

C 语言程序设计 **第二版**

C YUYAN CHENGXU SHEJI (DIERBAN)

刘克成 张凌晓 主编



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE



普通高等教育“十一五”国家级规划教材
普通高等教育计算机基础课程规划教材

C 语言程序设计

(第二版)

刘克成 张凌晓 主编



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书为普通高等教育“十一五”国家级规划教材,是根据教育部高等学校计算机基础课程教学指导委员会提出的“关于进一步加强高等学校计算机基础教学的意见”(简称白皮书)和《高等学校计算机基础教学战略研究报告暨计算机基础课程教学基本要求》中的“C语言程序设计”课程教学要求编写而成的。全书共分10章,系统地介绍了程序设计基本概念和结构化程序设计方法、C语言的各种数据类型、运算符与表达式、基本流程控制结构、模块化程序设计方法、数组、指针和文件系统,对面向对象编程基础也进行了初步阐述。

本书内容安排循序渐进,采取通俗易懂的讲解方法,并辅以丰富的例题和大量的习题强化关键知识点,程序调试和运行环境为Windows平台下的Visual C++ 6.0。另外,为帮助读者理解教材内容,强化动手能力,我们还针对每章内容编写了《C语言程序设计实验指导与习题解答(第二版)》,供学习时参考。

本书以面向工程实践和编程能力训练为主旨,以应用为目的。本书适合作为高等学校计算机程序设计教材;也可作为计算机程序设计培训教材,以及从事计算机程序设计的工程人员的参考书。

图书在版编目(CIP)数据

C语言程序设计 / 刘克成, 张凌晓主编. —2版—
北京: 中国铁道出版社, 2012. 1

普通高等教育“十一五”国家级规划教材. 普通高等教育
计算机基础课程规划教材
ISBN 978-7-113-13992-6

I. ①C… II. ①刘…②张… III. ①
C语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2011)第278076号

书 名: C语言程序设计(第二版)
作 者: 刘克成 张凌晓 主编

策 划: 吴宏伟
责任编辑: 杜 鹏 彭立辉
封面设计: 付 巍
封面制作: 白 雪
责任印制: 李 佳

读者热线: 400-668-0820

出版发行: 中国铁道出版社(100054,北京市西城区右安门西街8号)

网 址: <http://www.edusources.net>

印 刷: 北京昌平百善印刷厂

版 次: 2007年7月第1版 2012年1月第2版 2012年1月第7次印刷

开 本: 787mm×1092mm 1/16 印张: 18.5 字数: 440千

印 数: 27 001~31 000册

书 号: ISBN 978-7-113-13992-6

定 价: 32.00元

版权所有 侵权必究

凡购买铁道版图书,如有印制质量问题,请与本社教材图书营销部联系调换。电话:(010)63550836

打击盗版举报电话:(010)63549504

普通高等教育计算机基础课程规划教材

主任委员：冯博琴

副主任委员：管会生 李凤霞

委 员：（按姓氏笔画排序）

刘红梅 曲建民 李俊山

何东健 张长海 周 苏

唐 翔 高 飞 曹岳辉

编
审
委
员
会

计算机基础教学在我国高等教育中已有 30 多年的发展历史, 已经成为我国高等教育的重要组成部分, 是培养大学生综合素质的重要环节。计算机不仅为解决专业领域问题提供有效的方法和手段, 而且提供了一种独特的处理问题的思维方式; 计算机及互联网有着极其丰富的信息和知识资源, 为学生学习提供了广阔的空间以及良好的学习工具; 善于使用互联网和办公软件是良好的交流表达能力和团队合作能力的重要基础; 同时, 计算机基础教学也为学生创新能力的培养奠定了基础。不难发现, 现在几乎所有领域的重大成就无不得益于计算科学的支持, 计算科学已经和理论科学、实验科学并列成为推进社会文明进步和科技发展的三大手段。事实上, 当今任何一项被称为“高科技”的项目或专业、职业, 无一不是与计算机紧密结合的。计算机基础教学应致力于使大学生掌握计算科学的基本理论和方法, 为培养复合型创新人才服务。

本届教指委以科学发展观为指导, 为促进计算机基础教学不断向科学、规范、成熟的方向发展, 于 2009 年 10 月发布了《高等学校计算机基础教学战略研究报告暨计算机基础课程教学基本要求》(以下简称《基本要求》), 它充实了“4 个领域×3 个层次”的计算机基础教学的知识结构, 提出和构建了计算机基础教学的实验体系, 科学地描述各专业大类核心课程的教学基本要求。

《基本要求》提出了计算机基础教学应该达到的 4 项“能力结构”要求, 即对计算机的认知能力、利用计算机解决问题的能力、基于网络的协同能力、信息社会中的终身学习能力。以此为源头, 构建培养这 4 种能力的两大支柱, 即计算机基础教学的“知识体系”和“实验体系”。这两大体系中蕴含着计算机基础教学所包含的所有内容, 即 148 个知识单元、884 个知识点、119 个实验单元和 529 个技能点。根据教学目标, 可以从中选取若干知识单元、知识点、实验单元和技能点, 构建所需课程。这项研究基本上厘清了我国高校计算机基础教学的体系、内容和要求, 向科学、规范和可操作的方向迈出了一大步。

中国铁道出版社热心于计算机教育, 在计算机基础教学方面办了许多实事, 在高校师生中赢得了良好口碑。在《基本要求》发布之后, 我们组织国内一批知名教授和有实力的作者, 按照《基本要求》编写了本丛书, 以推动《基本要求》的贯彻, 提高高校计算机基础教学质量。

本丛书定位于应用型本科, 内容充分体现应用性, 兼顾基础性; 强调学生的动手能力培养, 避免过多的理论内容; 教材尽量采用案例驱动。丛书按照计算机基础教学六门核心课程组织, 有的课程或因平台不同, 或因教材编写风格、定位等不同, 会有一门课程多本教材的情况, 这是为了给老师提供更多的选择, 以使其找到更合适自己的优秀教材。

我们希望本丛书的出版, 能对推动我国高校的计算机基础教学改革尽到一份力量。书中难免存在不足之处, 恳望读者不吝指正。谢谢大家。

冯博琴

2010.10.8

冯博琴, 西安交通大学教授, 博士生导师, 现任教育部 2006—2010 年高校计算机基础课程教学指导委员会副主任委员, 全国计算机基础教育研究会副会长, 陕西省计算机教育研究会理事长。

无论在国内还是在国外，C 语言都是应用广泛、最具影响的程序设计语言之一。它概念简洁，数据类型丰富，运算符多，功能丰富，表达能力强，使用灵活，既有高级语言的优点，又具有低级语言的功能（能对硬件直接进行操作）。因此，它既适合编写应用程序，又适合编写系统程序。C 语言生成的目标程序执行效率高，具有良好的可移植性，是一种理想的结构化程序设计语言，多年来深受广大用户的喜爱。2007 年编写的《C 语言程序设计》出版后，被许多高校选用为教材，对本书给予了肯定和好评，我们根据教学实践和读者反馈的意见以及《高等学校计算机基础教学战略研究报告暨计算机基础课程教学基本要求》（2009）中的“C 语言程序设计”课程教学要求，对部分章节内容进行了调整、补充和重编形成第二版。

第一版以 ANSIC（美国国家标准局 1987 年公布 C 语言）为基础，书中的所有程序都是在 TC 2.0 环境下调试的，第二版以 C99（ISO 9899:1999）为基础，程序调试和运行环境为 Windows 平台下的 Visual C++ 6.0。本书在体系结构内容安排上，结合全国计算机等级考试二级考试的要求，采用实例导入和案例教学的方法，将难点分散，突出重点，充分体现算法设计和结构化程序设计思想。另外，每章都安排有习题，以便加强读者对所学关键知识的理解与掌握。

本书以面向工程实践和编程能力训练为主旨，以应用为背景，全面介绍了 C 语言的基本知识和结构化程序设计方法。全书共分 10 章：第 1 章介绍了 C 语言的初步知识；第 2 章介绍了 C 语言的数据类型、常量与变量和简单的输入/输出函数；第 3 章对 C 语言运算符和表达式进行了介绍；第 4 章综合介绍了 C 语言的基本流程控制结构、结构化程序设计的基本方法和有关算法的知识；第 5 章阐述了模块化程序设计方法；第 6 章讨论了 C 语言数组的定义与使用；第 7 章对指针的概念与应用进行了阐述；第 8 章重点介绍了结构体，并对共用体和枚举类型进行了简要介绍；第 9 章介绍了 C 语言的文件系统；第 10 章对面向对象程序设计的基本概念、特点及基本方法进行了初步阐述。

全书程序采用统一的代码规范编写，采用现代风格定义和声明函数，用传统流程图描述算法，所有程序都按照结构化程序设计方法编写，有助于学生养成良好的程序设计习惯。本书内容充实，循序渐进，选材上注意系统性、实用性。为配合读者学习，我们还编写了《C 语言程序设计实验指导与习题解答（第二版）》，与本书配套使用。

本书由南阳理工学院刘克成和张凌晓任主编，杨新锋、赵东江、鲁庆宾和孙晓莹任副主编。其中，第 1 章由刘克成编写，第 2 章和第 4 章由张凌晓编写，第 3 章和第 8 章由杨新锋编写，第 6 章和第 9 章由赵东江编写，第 5 章及附录由鲁庆宾编写，第 7 章和第 10 章由孙晓莹编写。

在本书的编写过程中，承蒙西安交通大学冯博琴教授和中国铁道出版社的热情支持与指导，在此表示衷心感谢。同时，我们在编写过程中还参阅了大量的网上资源和其他参考文献，在此对其作者和提供者一并表示感谢。

由于计算机科学技术发展迅速，程序设计的教学内容、方法和手段日新月异，且编者水平有限，书中难免有不足之处，敬请读者批评指正，以便再版时修改完善。

编者

2011 年 11 月

第一版前言

FOREWORD

C 语言是应用广泛、最具影响力的程序设计语言之一。它概念简洁,数据类型丰富,运算符多,表达能力强,使用灵活;它既有高级语言的优点,又具有低级语言的功能(能对硬件直接进行操作),因此既适合编写应用程序,又适合编写系统程序,并且生成的目标程序执行效率高,具有良好的可移植性,是一种理想的结构化程序设计语言,赢得了广大用户的喜爱。我们根据非计算机专业计算机基础课程教学指导分委员会提出的“关于进一步加强高等学校计算机基础教学的意见”(简称白皮书)中“高等学校非计算机专业计算机基础课程教学基本要求”部分的 C 语言程序设计课程大纲编写而成,其目的是让读者在没有任何编程基础的情况下就可以很容易地学习和掌握 C 语言的基本内容。本书在体系结构内容安排上将难点分散,使读者能循序渐进地逐步深入,并且每章都安排有习题,以便加强读者对所学关键知识点的理解与掌握。

本书全面介绍了 C 语言的基本知识和结构化程序设计方法。本书共有 10 章,第 1 章介绍了程序设计的基本知识,第 2 章介绍了 C 语言的数据类型、常量与变量和简单的输入输出函数,第 3 章对 C 语言运算符和表达式进行了介绍,第 4 章综合介绍了 C 语言的顺序结构、选择结构和循环结构等进行结构化程序设计的基本方法和有关算法的知识,第 5 章阐述了模块化程序设计方法,第 6 章讨论了 C 语言数组的定义与使用,第 7 章对指针的概念与应用作了充分的阐述,第 8 章重点介绍了结构体,并对共用体和枚举类型作了简要介绍,第 9 章介绍了 C 语言的文件系统,第 10 章对面向对象编程的基本概念、特点及面向对象程序设计的基本方法进行了初步阐述。

本书是作者在多年从事 C 语言程序设计的教学实践基础上,结合讲义编写而成的。本书采用现代风格定义和声明函数,用传统流程图描述算法,所有程序都按照结构化程序设计方法编写,有助于养成良好的程序设计习惯。本书内容充实,循序渐进,选材上注意系统性、实用性,例题可以在 Turbo C 2.0 或 Visual C++ 6.0 集成开发环境下编译运行,并且为配合读者学习,我们另编写了《C 语言程序设计实验指导与习题解答》一书,作为本书的配套教材。

本书由刘克成教授主编,张凌晓、邵艳玲、杨彩霞为副主编。第 7 章由刘克成编写;第 8 章及附录由张凌晓编写;第 5、10 章由邵艳玲编写;第 1、6 章由张晓民编写,第 4 章由杨新锋编写;第 2、3、9 章由杨彩霞编写。全书总纂工作由刘克成、张凌晓和邵艳玲负责完成。

在本书的编写过程中,承蒙西安交通大学冯博琴教授的热情支持与指导,在此表示衷心感谢!同时,又参阅了大量的网上资源和其他参考文献,在此对它们的作者和提供者一并表示感谢。

由于计算机科学技术发展迅速,程序设计的教学内容、方法和手段日新月异,加以编者水平有限,书中难免有不足之处,敬请读者批评指正,以便再版时修改完善。

编者

2005 年 11 月

| | |
|---------------------------------------|----|
| 第 1 章 C 语言程序设计概述 | 1 |
| 1.1 计算机程序 | 1 |
| 1.2 计算机语言 | 1 |
| 1.3 程序设计 | 3 |
| 1.3.1 程序设计定义 | 3 |
| 1.3.2 程序设计基本步骤 | 3 |
| 1.3.3 C 程序设计的开发过程 | 4 |
| 1.4 C 语言的发展及其特性 | 4 |
| 1.4.1 C 语言的发展过程 | 4 |
| 1.4.2 C 语言的特性 | 5 |
| 1.5 C 语言程序组成结构 | 6 |
| 1.5.1 C 语言程序举例 | 6 |
| 1.5.2 C 语言程序组成结构的特点 | 8 |
| 1.5.3 C 语言字符集 | 8 |
| 1.5.4 C 语言标识符 | 9 |
| 1.6 用 Visual C++ 6.0 运行 C 程序的方法 | 10 |
| 1.6.1 Visual C++ 6.0 的安装和启动 | 10 |
| 1.6.2 建立和运行一个文件的程序 | 10 |
| 1.7 怎样学习 C 程序设计 | 13 |
| 1.7.1 为什么要学习 C 语言 | 13 |
| 1.7.2 学习 C 程序设计的方法 | 14 |
| 小结 | 14 |
| 习题 | 15 |
| 第 2 章 数据类型与简单输入/输出 | 17 |
| 2.1 C 语言基本数据类型 | 17 |
| 2.1.1 数据类型概述 | 17 |
| 2.1.2 整数类型 | 18 |
| 2.1.3 实数类型 | 19 |
| 2.1.4 字符类型 | 20 |
| 2.2 常量与变量 | 20 |
| 2.2.1 常量 | 20 |
| 2.2.2 变量 | 23 |
| 2.3 数据的输入与输出 | 25 |
| 2.3.1 数据的输出 | 25 |
| 2.3.2 数据的输入 | 30 |
| 小结 | 34 |
| 习题 | 35 |

| | |
|------------------------------|-----------|
| 第3章 运算符与表达式 | 38 |
| 3.1 基本运算符及其表达式..... | 38 |
| 3.1.1 C运算符概述..... | 38 |
| 3.1.2 算术运算符及其表达式..... | 39 |
| 3.1.3 赋值运算符及其表达式..... | 41 |
| 3.1.4 关系运算符及其表达式..... | 43 |
| 3.1.5 逻辑运算符及其表达式..... | 44 |
| 3.1.6 位运算符及其表达式..... | 46 |
| 3.1.7 条件运算符及其表达式..... | 47 |
| 3.1.8 逗号运算符和 sizeof 运算符..... | 48 |
| 3.2 混合运算时数据类型的转换..... | 49 |
| 3.2.1 类型隐式转换..... | 49 |
| 3.2.2 类型显式转换..... | 50 |
| 小结..... | 51 |
| 习题..... | 52 |
| 第4章 程序流程控制 | 56 |
| 4.1 算法..... | 56 |
| 4.1.1 算法的概念..... | 56 |
| 4.1.2 算法的组成要素..... | 57 |
| 4.1.3 算法的描述..... | 58 |
| 4.2 C语言的基本语句..... | 60 |
| 4.3 顺序结构..... | 61 |
| 4.4 选择结构..... | 63 |
| 4.4.1 if...else 语句..... | 63 |
| 4.4.2 if 语句..... | 65 |
| 4.4.3 else if 语句..... | 66 |
| 4.4.4 if 语句的嵌套..... | 67 |
| 4.4.5 switch 语句结构..... | 71 |
| 4.5 循环结构..... | 74 |
| 4.5.1 穷举与迭代算法..... | 76 |
| 4.5.2 while 语句..... | 77 |
| 4.5.3 do...while 语句..... | 81 |
| 4.5.4 for 语句..... | 81 |
| 4.5.5 流程转向语句..... | 83 |
| 4.5.6 循环结构的嵌套..... | 84 |
| 小结..... | 88 |
| 习题..... | 89 |

| | |
|--------------------------|------------|
| 第 5 章 模块化编程 | 96 |
| 5.1 模块化设计与函数..... | 96 |
| 5.1.1 函数的定义 | 97 |
| 5.1.2 函数的调用 | 98 |
| 5.1.3 函数的参数传递..... | 101 |
| 5.1.4 函数的嵌套调用..... | 102 |
| 5.1.5 函数的递归调用..... | 103 |
| 5.1.6 系统标准函数..... | 106 |
| 5.2 局部变量和全局变量..... | 107 |
| 5.2.1 局部变量 | 107 |
| 5.2.2 全局变量 | 109 |
| 5.3 变量的存储属性 | 111 |
| 5.3.1 自动变量 | 112 |
| 5.3.2 静态变量 | 114 |
| 5.3.3 外部变量 | 115 |
| 5.3.4 寄存器变量 | 116 |
| 5.4 内部函数和外部函数..... | 117 |
| 5.4.1 内部函数 | 117 |
| 5.4.2 外部函数..... | 117 |
| 5.5 编译预处理 | 117 |
| 5.5.1 文件包含 | 117 |
| 5.5.2 宏定义 | 118 |
| 5.5.3 条件编译 | 121 |
| 5.6 函数综合应用举例..... | 123 |
| 小结..... | 125 |
| 习题..... | 126 |
| 第 6 章 数组 | 130 |
| 6.1 一维数组 | 130 |
| 6.1.1 一维数组的定义..... | 130 |
| 6.1.2 一维数组元素的引用 | 131 |
| 6.1.3 一维数组的内存映像 | 131 |
| 6.1.4 一维数组的初始化..... | 132 |
| 6.1.5 一维数组的程序举例..... | 132 |
| 6.2 二维数组 | 134 |
| 6.2.1 二维数组的定义..... | 134 |
| 6.2.2 二维数组的引用..... | 134 |
| 6.2.3 二维数组的内存映像..... | 135 |
| 6.2.4 二维数组的初始化 | 136 |

| | | |
|------------|---------------------------|------------|
| 6.2.5 | 二维数组的程序举例 | 137 |
| 6.3 | 字符数组与字符串 | 139 |
| 6.3.1 | 一维字符数组 | 139 |
| 6.3.2 | 字符串 | 140 |
| 6.3.3 | 二维字符数组 | 141 |
| 6.3.4 | 常用的字符串处理库函数 | 142 |
| 6.3.5 | 字符数组程序举例 | 145 |
| 6.4 | 数组作为函数的参数 | 146 |
| 6.4.1 | 数组元素只能作函数实参 | 147 |
| 6.4.2 | 数组名作函数参数 | 147 |
| 6.5 | 数组综合应用 | 150 |
| 小结 | | 152 |
| 习题 | | 153 |
| 第7章 | 指针 | 157 |
| 7.1 | 指针的概念 | 157 |
| 7.1.1 | 地址与指针 | 157 |
| 7.1.2 | 指针变量 | 158 |
| 7.1.3 | 指针运算 | 158 |
| 7.2 | 指针与变量 | 161 |
| 7.3 | 指针与数组 | 162 |
| 7.3.1 | 指针和一维数组 | 162 |
| 7.3.2 | 指针和二维数组 | 165 |
| 7.3.3 | 指针数组 | 167 |
| 7.4 | 指针与字符串 | 168 |
| 7.4.1 | 字符指针与字符数组 | 168 |
| 7.4.2 | 字符指针数组 | 170 |
| 7.5 | 指针与函数 | 172 |
| 7.5.1 | 指针作为函数形式参数 | 172 |
| 7.5.2 | 返回指针的函数 | 175 |
| 7.5.3 | 指向函数的指针 | 176 |
| 7.6 | 指向指针的指针 | 178 |
| 7.7 | 命令行参数 | 180 |
| 7.8 | 指针综合应用举例 | 181 |
| 小结 | | 184 |
| 习题 | | 184 |
| 第8章 | 结构体、共用体与枚举类型 | 190 |
| 8.1 | 结构体 | 190 |
| 8.1.1 | 结构体类型的定义 | 190 |

| | | |
|---------------|-------------------------------|------------|
| 8.1.2 | 结构体类型变量 | 191 |
| 8.1.3 | 结构体类型数组 | 196 |
| 8.1.4 | 结构体类型指针 | 199 |
| 8.1.5 | 结构体与函数 | 202 |
| 8.1.6 | 动态内存分配 | 205 |
| 8.2 | 共用体 | 208 |
| 8.3 | 位段结构 | 213 |
| 8.4 | 枚举类型 | 214 |
| 8.5 | 用 typedef 定义类型 | 217 |
| 8.6 | 结构体应用举例 | 218 |
| | 小结 | 225 |
| | 习题 | 226 |
| 第 9 章 | 文件 | 231 |
| 9.1 | 文件概述 | 231 |
| 9.1.1 | 文件的概念 | 231 |
| 9.1.2 | C 语言的文件系统 | 232 |
| 9.2 | 标准文件操作 | 232 |
| 9.2.1 | 文件类型指针 | 233 |
| 9.2.2 | 文件的打开 | 233 |
| 9.2.3 | 文件的关闭 | 235 |
| 9.2.4 | 文件的顺序读/写 | 235 |
| 9.2.5 | 文件的随机读/写 | 243 |
| 9.2.6 | 文件检测 | 245 |
| | 小结 | 246 |
| | 习题 | 246 |
| 第 10 章 | 面向对象程序设计 | 251 |
| 10.1 | 结构化程序设计与面向对象程序设计 | 251 |
| 10.1.1 | 结构化程序设计与面向对象程序设计概述 | 251 |
| 10.1.2 | 结构化程序设计和面向对象程序设计解决问题的比较 | 252 |
| 10.2 | 面向对象程序设计的基本概念 | 254 |
| 10.3 | 面向对象程序设计的基本特点 | 255 |
| 10.4 | 类和对象 | 256 |
| 10.4.1 | 类的定义 | 256 |
| 10.4.2 | 对象的定义 | 257 |
| 10.4.3 | 对象的初始化 | 257 |
| 10.5 | 继承和派生 | 259 |
| 10.5.1 | 派生类的定义 | 259 |
| 10.5.2 | 派生类程序举例 | 260 |

| | |
|-------------------------|-----|
| 10.6 多态性 | 261 |
| 10.6.1 函数重载 | 261 |
| 10.6.2 运算符重载 | 262 |
| 10.6.3 虚函数 | 263 |
| 10.7 面向对象的软件开发 | 264 |
| 小结 | 266 |
| 习题 | 266 |
| 附录 A ASCII 码表 | 269 |
| 附录 B 关键字 | 271 |
| 附录 C 运算符的优先级和结合方向 | 272 |
| 附录 D 常用库函数 | 273 |
| 附录 E C 语言常见出错信息 | 278 |
| 参考文献 | 282 |

第 1 章 C 语言程序设计概述

在学习任何程序设计之前，需要先了解一些基础性的知识。对于初学者，本章可作为快速入门内容；对于有程序设计基础者，可以略过本章，进而学习后续章节。本章主要包括程序设计的基本知识：计算机程序与计算机语言；程序设计的步骤和方法；C 语言的发展及其特性；通过简单的 C 语言例子，介绍了 C 语言程序的组成结构；详细介绍用 Visual C++ 6.0 运行 C 程序的步骤与方法；最后，给出学习 C 语言的建议和方法。

教学目的：

- 理解计算机程序的概念；
- 了解程序设计语言的发展；
- 了解程序设计的基本步骤；
- 掌握 Visual C++ 6.0 程序调试的方法；
- 了解学习 C 语言的重要性。

1.1 计算机程序

人与计算机不能直接互动和交流，如果想操纵计算机就必须提前安排好各项任务并转化成它能识别的东西，只有这样计算机才会自动进行所有的工作。其实，计算机的每一个操作都是根据人们事先指定的指令进行的。

计算机程序就是一组计算机能识别和执行的指令。每一条指令使计算机执行特定的操作，完成相应的工作。只要让计算机执行这个程序，计算机就会“自动地”执行各条指令，有条不紊地进行工作。为了使计算机系统能实现各种功能，需要成千上万个程序。这些程序大多数是由计算机软件设计人员根据需要设计好的，作为计算机软件系统的一部分提供给用户使用。此外，用户还可以根据自己的实际需要设计一些应用程序，例如学生成绩管理程序、财务管理程序、工程中的计算程序等。

总之，计算机的一切操作都是由程序控制的，离开程序，计算机将一事无成。所以，计算机的本质是程序的机器，程序和指令是计算机系统中最基本的概念。只有懂得程序设计，才能真正了解计算机是怎样工作的，才能更深入地使用计算机。

1.2 计算机语言

在世界各国的交流中，不同国家的人使用不同的语言，如果他们之间想交流就需要“翻译”。“翻译”的职能就是懂得双方的语言，起到信息相互沟通的作用。人与计算机之间想进行交流，也需要“翻译”，即人与计算机都能识别的语言，这就是计算机语言。计算机语言一般分为 4 类：低级语言（机器语言、汇

编语言)、高级语言 (C、C++、Java 等)、专用语言 (例如: CAD 系统中的绘图语言和 DBMS 中的数据库查询语言) 和脚本语言 (例如: JavaScript、Ruby、Python 等)。计算机语言经历了以下几个发展阶段:

1. 机器语言

机器指令就是计算机能直接识别和接受的二进制代码 (即“0”和“1”序列)。机器指令的集合就是该计算机的机器语言, 它是计算机唯一能够直接识别和执行的语言。

因此, 用机器语言编写的程序就是一个二进制文件。一条机器语言称为一条指令, 指令是不可分割的最小功能单元。而且, 由于每台计算机的指令系统往往各不相同, 所以, 在不同计算机上执行同一程序, 就必须编写不同版本, 这样就造成了重复工作。但由于使用的是针对特定型号计算机的语言, 故而运算效率是所有语言中最高的。机器语言是第一代计算机语言。

2. 汇编语言

汇编语言是人们用一些英文字母和数字表示一条指令。例如, 用 ADD 代表加法, MOV 代表数据传递, 等等。这样一来, 人们很容易读懂并理解程序在干什么, 纠错及维护都变得方便了。像这种计算机并不能直接识别和执行的符号语言的指令, 需要用一种称为汇编程序的软件, 把符号语言的指令转换为机器指令。机器指令转换的过程称为“代真”或“汇编”, 因此, 汇编语言又称为符号语言。

不同型号计算机的机器语言和汇编语言是互不通用的。用甲机器的机器语言编写的程序在乙机器上不能使用。机器语言和汇编语言是完全依赖于具体机器特性的, 其可读性和可移植性都很差, 是面向机器的语言。由于它“贴近”计算机, 或者说离计算机“很近”, 称为计算机低级语言。

3. 高级语言

高级语言的出现, 是为了克服低级语言的缺点, 它很接近于人们习惯使用的自然语言和数学语言。程序中用到的语句和指令是用英文单词表示的, 程序中所用的运算符、运算表达式和人们日常所用的数学式子差不多, 很容易理解。这种语言功能很强, 且不依赖于具体机器, 用它写出的程序对任何型号的计算机都适用 (或只须作很少的修改), 它与具体机器距离较远, 故称为计算机高级语言。

当然, 计算机也是不能直接识别高级语言程序的, 也要进行“翻译”。用一种称为编译程序的软件把用高级语言编写的程序 (称为源程序, Source Program) 转换为机器指令的程序 (称为目标程序, Object Program), 然后让计算机执行机器指令程序, 最后得到结果。

高级语言经历了不同的发展阶段:

1954 年, 第一个完全脱离机器硬件的高级语言 FORTRAN 问世; 1969 年, 提出了结构化程序设计方法; 1970 年, 第一个结构化程序设计语言 Pascal 语言出现, 标志着结构化程序设计时期的开始。20 世纪 80 年代初开始, 在软件设计思想上, 又产生了一次革命, 其成果就是面向对象的程序设计。在此之前的高级语言, 几乎都是面向过程的, 程序的执行是流水线似的, 在一个模块被执行完成前, 人们不能干别的事, 也无法动态地改变程序的执行方向。像 C++、Java、Visual Basic、Delphi 等就是典型的面向对象程序设计语言的代表。

(1) 非结构化的语言。初期的语言属于非结构化的语言, 编程风格比较随意, 只要符合语法规则即可, 没有严格的规范要求, 程序中的流程可以随意跳转。人们往往追求程序执行的效率而采用了许多“小技巧”, 使程序变得难以阅读和维护。早期的 BASIC、FORTRAN 和 ALGOL 等都属于非结构化的语言。

(2) 结构化语言。为了解决以上问题, 提出了“结构化程序设计方法”, 规定程序必须由具有良好特性的基本结构 (顺序结构、分支结构、循环结构) 构成, 程序中的流程不允许随意跳转,

程序总是由上而下顺序执行各个基本结构。这种程序结构清晰,易于编写、阅读和维护。QBASIC、FORTRAN 77 和 C 语言等属于结构化的语言,这些语言的特点是支持结构化程序设计方法。

以上两种语言都是基于过程的语言,在编写程序时需要具体指定每一个过程的细节。在编写规模较小的程序时,还能得心应手,但在处理规模较大的程序时,就显得捉襟见肘、力不从心了。

(3) 面向对象的语言。在实践的发展中,人们又提出了面向对象的程序设计方法。程序面对的不是过程的细节,而是一个个对象,对象是由数据以及对数据进行的操作组成的。10 多年以来,在处理规模较大的问题时,开始使用面向对象的语言。C++、C#、Visual Basic 和 Java 等语言是支持面向对象程序设计方法的语言。

(4) 面向应用的语言。面向应用的语言也是高级语言的下一个发展目标,也就是说:只需要告诉程序你要干什么,程序就能自动生成算法,自动进行处理,这也是非过程化的程序语言。

进行程序设计时,必须要用到计算机语言,人们根据任务的需要选择合适的语言,编写出程序,然后运行程序得到结果。

1.3 程序设计

1.3.1 程序设计定义

对复杂程度较高的问题,想直接编写程序是不现实的,必须从问题描述入手,经过对解题算法的分析、设计直至程序的编写、调试和运行等一系列过程,最终得到能够解决问题的计算机应用程序,此过程称为程序设计,简单地称计算机编程。在程序设计中,程序员需要了解各种开发语言和开发平台的优缺点,并且懂得如何根据问题的大小和难易程度选择最合适的开发工具。由于现在的实用程序越来越大,在大多数情况下,单用一种开发语言和开发平台已不能解决问题,需要多种开发工具的通力协作。

1.3.2 程序设计基本步骤

简单的程序设计,具体步骤如下:

(1) 问题分析与方案确定:程序将以数据处理的方式解决客观世界中的问题,因此在程序设计之初,首先应该将实际问题用数学语言描述出来,形成一个抽象的、具有一般性的数学问题,从而给出问题的抽象数学模型,然后制定解决该模型所代表数学问题的算法。数学模型精确地阐述了模型本身所涉及的各种概念、已知条件、所求结果,以及已知条件与所求结果之间的联系等各方面的信息。数学模型是进一步确定解决所代表数学问题的算法的基础。数学模型和算法的结合将给出问题的解决方案。

(2) 算法描述:具体的解决方案确定后,需要对所采用的算法进行描述,算法的初步描述可以采用自然语言方式,然后逐步将其转化为程序流程图或其他直观方式。这些描述方式比较简单明确,能够比较明显地展示程序设计思想,是进行程序调试的重要参考。

(3) 编写程序:使用计算机系统提供的某种程序设计语言,根据上述算法进行描述,将已设计好的算法表达出来。使得非形式化的算法转变为形式化的由程序设计语言表达的算法,这个过程称为程序编制(编码)。程序的编写过程需要反复调试才能得到可以运行且结果“正确”的程序。

(4) 程序测试:程序编写完成后必须经过科学的、严格的运行和测试,才能最大限度地保证

程序的正确性。同时,通过测试可以对程序的性能作出评估。

(5) 编写文档:程序文档一般包括程序名称、功能、运行环境、程序的装入和启动、需要输入的数据,以及使用的注意事项等。软件是计算机程序和程序文档的总称。

通过上述过程可以看到,程序设计过程是算法、数据结构和程序设计语言相统一的过程。问题和算法的最初描述,无论是描述形式上还是内容上,离最终以计算机语言描述的算法(程序)还有相当大的差距。对于一个规模不是很大的问题,程序设计的核心是算法设计和数据结构设计,只要成功地构造出解决问题的高效算法和数据结构,则完成剩下的任务就不存在太大困难。

1.3.3 C 程序设计的开发过程

1. 设计算法

针对具体的问题,分析、建立解决问题的物理或数学模型,并将解决方法采用某种方式描述出来,为进行 C 语言实际编程打下良好基础。

2. 编辑

使用一个文本编辑器编辑 C 语言源程序,并将其保存为文件扩展名为“.c”的文件。

3. 编译

编译就是将编辑好的 C 语言源程序翻译成二进制目标代码的过程。编译过程由 C 语言编译系统自动完成。编译时首先检查源程序的每一条语句的语法错误,当发现错误时,就在屏幕上显示错误的位置和错误类型信息。此时,要再次调用编辑器进行查错并修改,然后再进行编译,直到排除所有的语言和语义错误。正确的源程序文件经过编译后,在磁盘上生成同名的目标文件(.obj)。

4. 连接

将目标文件和库函数等连接在一起形成一个扩展名为“.EXE”的可执行文件。如果函数名称写错或漏写包含库函数的头文件,则可能提示错误信息,从而得到程序错误数据。

5. 运行

可以脱离 C 语言编译系统,直接在操作系统下运行。若运行程序后达到预期的目的,则 C 程序的开发工作到此完成,否则要进一步修改源程序,重复编辑—编译—连接—运行的过程,直到取得正确结果为止。C 语言程序上机过程如图 1-1 所示。

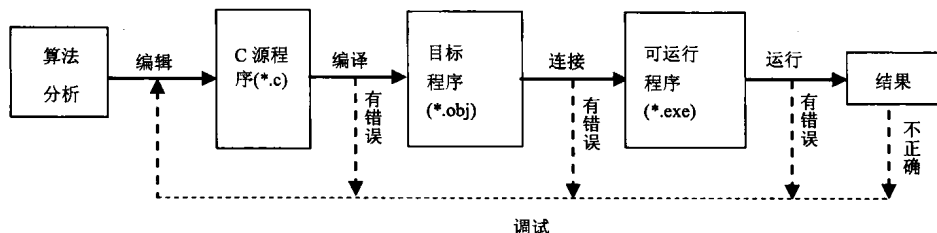


图 1-1 C 语言程序上机过程

1.4 C 语言的发展及其特性

1.4.1 C 语言的发展过程

1967年,英国剑桥大学的 Martin Richards 推出了没有类型的 BCPL(Basic Combined Programming