



普通高等教育“十一五”国家级规划教材

西安交通大学“十一五”规划教材

# 程序设计与C语言

(第3版)

梁力 原盛 编著



西安交通大学出版社  
XI'AN JIAOTONG UNIVERSITY PRESS



普通高等教育“十一五”国家级规划教材

西安交通大学“十一五”规划教材

# 程序设计与C语言

## (第3版)

梁力 原盛 编著

|                        |        |            |
|------------------------|--------|------------|
| ISBN 978-7-309-07100-9 | 30.00元 | 2010年8月第3版 |
| ISBN 978-7-309-07100-9 | 30.00元 | 2010年8月第3版 |

西安交通大学出版社  
XI'AN JIAOTONG UNIVERSITY PRESS

地址：西安交通大学

## 内 容 简 介

本教材以程序设计方法为主线,以 C 语言作为典型的程序设计语言,全面系统地介绍了结构化程序设计的思想和方法。教材中,通过将例子分成三个层次的方法,详细讲述了 C 语言的基本概念、语法规则和语义特点,以及在 C 语言的环境下,编写程序的思路、方法和技巧。

本教材语言通俗易懂,内容深入浅出,重点突出,范例程序丰富,实用性、技巧性强,强调动手实践,不仅可以作为大专院校本科生的教材,也可以供从事计算机、自动化和相关领域的科研人员参考自学。

---

### 图书在版编目(CIP)数据

程序设计与 C 语言/梁力,原盛编著. —3 版:西安:  
西安交通大学出版社,2010.8  
ISBN 978-7-5605-3603-3

I. ①程… II. ①梁… ②原… III. ①C 语言-程序设计-教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2010)第 105305 号

---

书 名 程序设计与 C 语言(第 3 版)  
编 著 梁 力 原 盛  
责任编辑 屈晓燕 贺峰涛

---

出版发行 西安交通大学出版社  
(西安市兴庆南路 10 号 邮政编码 710049)  
网 址 <http://www.xjtupress.com>  
电 话 (029)82668357 82667874(发行中心)  
(029)82668315 82669096(总编办)  
传 真 (029)82668280  
印 刷 陕西新世纪印刷厂

---

开 本 787mm×1092mm 1/16 印张 21.5 字数 521 千字  
版次印次 2010 年 8 月第 3 版 2010 年 8 月第 1 次印刷  
书 号 ISBN 978-7-5605-3603-3/TP·531  
定 价 30.00 元

---

读者购书、书店添货,如发现印装质量问题,请与本社发行中心联系、调换。

订购热线:(029)82665248 (029)82665249

投稿热线:(029)82664954

读者信箱:jdlgy@yahoo.cn

版权所有 侵权必究

# 前 言

本书是一本讲授程序设计和程序设计语言的教科书。

程序设计是学习计算机知识非常重要的基础课程之一,是学好计算机系列课程的基础。程序设计课程包括两方面的内容:程序设计方法和程序设计语言。自从第一台计算机诞生以来,程序设计方法与程序设计语言就一直不断地发展,从主要用于科学计算的程序设计,到20世纪70年代的结构化程序设计,进而到80年代的面向对象的程序设计,其目的是为了使用计算机这一二十世纪最伟大的科研成果,人类智慧的结晶能够更好地为人类服务。

程序设计语言也从机器语言、汇编语言、高级语言,到目前更加便捷,更加拟人化的各种可视化语言。这些都使人们感到了计算机科学的飞速发展和它的勃勃生机。

编写本教材的目的是为学生打下一个扎扎实实的程序设计的基本理论与基本方法,使学生熟练掌握一门典型的程序设计语言,以适应计算机科学不断推出的新方法、新工具。书中以介绍程序设计方法为主,结合一种典型的程序设计语言,通过列举大量的应用实例系统地,较为全面地介绍结构化程序设计的思想和方法。

学习程序设计的目的是建立程序设计的基本思想,掌握程序设计的基本方法,不可能只通过一门程序设计课程的学习就要求学生立即编写大型软件。要想成为一名合格的程序设计者,一定要掌握了许多计算机理论,还有不断的实践,不断的积累经验。任何人不可能只靠在学校学习的一门语言包打天下。只有掌握了程序设计的基本理论和基本方法,才能适应计算机不断发展的需要。

作者长期从事程序设计基础课程的教学工作,本教材是作者长期教学实践的总结。编写教材过程中,作者研究了学生学习程序设计认识规律,采用了通俗易懂的语言,循序渐进的方法。本教材的特点如下:

1. 以程序设计方法为纲,较为系统全面地介绍了程序设计的发展和结构化程序设计,其目的让读者掌握程序设计的基本理论和基本方法。
2. 以C语言作为典型的程序设计语言,介绍了用程序设计语言描述结构化程序设计方法。按结构化程序设计方法的三种基本结构介绍C语言的控制语句和程序设计。
3. 各章中列举了丰富的例子。例子分为三个层次,第1层次,帮助读者理解基本语法;第2层次,介绍基本的程序设计方法;第3个层次,综合举一些具有较强编程技巧和实用性的例子。教师可以根据教学实际需要和不同程度的学生选取例子。
4. 本书各章精选了典型习题,供学生选作。程序设计是一门实践性很强的课程,要求学生大量阅读程序,动手编写程序,上机调试程序,从实践中不断总结、积累经验。
5. 书中的例子都已上机检验。
6. 本书既可以作为计算机专业学生的教材,又可以作为非计算机专业学生的教材。

本书由梁力主编、统稿。原盛编写了第1、2、7、6、9章,梁力编写了第3、4、5、8章。西安交

通大学电信学院副院长董渭清教授仔细审阅了书稿,并提出了宝贵意见。在编写本教材过程中受到了西安交通大学软件学院副院长曾明教授、计算机应用研究所所长陈建明副教授的大力支持。多位资深教授围绕本书内容对如何搞好程序设计基础教学提出了许多建设性意见。韩菁老师在编书过程中给予了我们热情的帮助,提供了许多有意义的程序例子,丰富了本书的内容。在使用本教材的过程中,唐亚哲老师、钱屹老师提出许多具体的意见。在这里一并表示感谢。本书经过几届西安交通大学电信学院和软件学院本科生使用,编者在原书稿的基础上进行了修改。由于编者水平有限,书中肯定会有不少缺点和不足,恳请广大读者批评指正。

编 者

2010.2.1

# 目 录

## 前言

|                                       |      |
|---------------------------------------|------|
| <b>第 1 章 程序设计基础</b> .....             | (1)  |
| 1.1 计算机基础 .....                       | (1)  |
| 1.1.1 计算机硬件系统 .....                   | (1)  |
| 1.1.2 计算机软件系统 .....                   | (2)  |
| 1.1.3 计算机的发展 .....                    | (3)  |
| 1.1.4 计算机的发展方向 .....                  | (4)  |
| 1.2 程序设计基础 .....                      | (5)  |
| 1.2.1 程序及算法 .....                     | (6)  |
| 1.2.2 算法的特征和描述 .....                  | (7)  |
| 1.2.3 算法与程序设计 .....                   | (9)  |
| 1.2.4 程序设计语言 .....                    | (13) |
| 1.2.5 C 语言 .....                      | (14) |
| 1.3 程序设计发展史 .....                     | (15) |
| 1.4 结构化程序设计 .....                     | (16) |
| 1.4.1 结构化程序设计的发展 .....                | (16) |
| 1.4.2 结构化程序设计的特征与风格 .....             | (17) |
| 1.4.3 C 语言与结构化程序设计 .....              | (18) |
| 1.4.4 C 程序的执行 .....                   | (21) |
| 1.5 面向对象程序设计 .....                    | (22) |
| 1.5.1 面向对象程序设计的发展及基本概念 .....          | (22) |
| 1.5.2 面向对象程序设计的特征 .....               | (24) |
| 习题 .....                              | (25) |
| <b>第 2 章 常量、变量、数据类型、运算符和表达式</b> ..... | (26) |
| 2.1 常量和变量 .....                       | (26) |
| 2.1.1 标识符与关键字 .....                   | (26) |
| 2.1.2 常量和变量 .....                     | (27) |
| 2.2 数据类型 .....                        | (28) |
| 2.2.1 整型数据 .....                      | (28) |
| 2.2.2 实型数据 .....                      | (31) |
| 2.2.3 字符型数据 .....                     | (32) |
| 2.3 运算符 .....                         | (35) |

|       |             |      |
|-------|-------------|------|
| 2.3.1 | 算术运算符       | (35) |
| 2.3.2 | 自增、自减运算符    | (36) |
| 2.3.3 | 赋值运算符       | (38) |
| 2.3.4 | 关系运算符       | (39) |
| 2.3.5 | 逻辑运算符       | (40) |
| 2.3.6 | 逗号运算符       | (40) |
| 2.3.7 | 运算符的优先级和结合性 | (40) |
| 2.4   | 表达式         | (41) |
| 2.4.1 | 算术表达式       | (41) |
| 2.4.2 | 赋值表达式       | (42) |
| 2.4.3 | 关系表达式       | (43) |
| 2.4.4 | 逻辑表达式       | (44) |
| 2.4.5 | 逗号表达式       | (46) |
| 2.4.6 | 条件表达式       | (47) |
| 2.5   | 数据类型转换      | (48) |
| 2.5.1 | 自动类型转换      | (48) |
| 2.5.2 | 强制类型转换      | (49) |
| 2.6   | 位运算         | (51) |
|       | 习题          | (54) |

### 第3章 C语言程序控制语句与结构化程序设计的三种基本结构 (57)

|       |            |      |
|-------|------------|------|
| 3.1   | C语句概述      | (57) |
| 3.2   | 顺序结构程序设计   | (58) |
| 3.2.1 | 表达式语句      | (58) |
| 3.2.2 | 数据的输出      | (59) |
| 3.2.3 | 数据的输入      | (65) |
| 3.2.4 | 复合语句       | (71) |
| 3.2.5 | 顺序程序设计     | (72) |
| 3.3   | 分支程序设计     | (74) |
| 3.3.1 | if语句       | (74) |
| 3.3.2 | switch语句   | (76) |
| 3.3.3 | break语句    | (78) |
| 3.3.4 | 条件运算符      | (79) |
| 3.3.5 | 分支程序设计     | (80) |
| 3.4   | 循环程序设计     | (84) |
| 3.4.1 | for语句      | (84) |
| 3.4.2 | while语句    | (88) |
| 3.4.3 | do-while语句 | (90) |
| 3.4.4 | 循环嵌套       | (93) |

|              |                |       |
|--------------|----------------|-------|
| 3.4.5        | continue 语句    | (96)  |
| 3.4.6        | break 语句的进一步说明 | (98)  |
| 3.4.7        | 循环程序设计         | (99)  |
| 3.5          | 综合举例           | (102) |
|              | 习题             | (104) |
| <b>第 4 章</b> | <b>数组</b>      | (107) |
| 4.1          | 数组概述           | (107) |
| 4.2          | 一维数组           | (107) |
| 4.2.1        | 一维数组的定义        | (107) |
| 4.2.2        | 一维数组的存储结构      | (108) |
| 4.2.3        | 一维数组的引用        | (108) |
| 4.2.4        | 一维数组的输入输出      | (109) |
| 4.2.5        | 一维数组的初始化       | (110) |
| 4.2.6        | 一维数组程序举例       | (111) |
| 4.3          | 二维数组           | (118) |
| 4.3.1        | 二维数组的定义        | (118) |
| 4.3.2        | 二维数组的引用        | (120) |
| 4.3.3        | 二维数组的初始化       | (121) |
| 4.3.4        | 二维数组程序举例       | (121) |
| 4.4          | 字符数组           | (125) |
| 4.4.1        | 字符数组的定义        | (126) |
| 4.4.2        | 字符数组的初始化       | (126) |
| 4.4.3        | 字符数组的引用        | (128) |
| 4.4.4        | 字符数组的输入输出      | (128) |
| 4.4.5        | 字符串处理函数        | (129) |
| 4.4.6        | 程序举例           | (132) |
| 4.5          | 数组应用综合举例       | (134) |
|              | 习题             | (139) |
| <b>第 5 章</b> | <b>函数</b>      | (144) |
| 5.1          | 函数概述           | (144) |
| 5.2          | 函数定义           | (146) |
| 5.2.1        | 函数的定义形式        | (146) |
| 5.2.2        | 空函数            | (148) |
| 5.3          | 函数参数与函数的返回值    | (148) |
| 5.3.1        | 形式参数与实在参数      | (148) |
| 5.3.2        | 函数的返回值         | (150) |
| 5.4          | 函数的调用          | (150) |



|       |              |       |
|-------|--------------|-------|
| 5.4.1 | 函数调用         | (150) |
| 5.4.2 | 函数调用规则       | (151) |
| 5.5   | 函数的嵌套调用和递归调用 | (154) |
| 5.5.1 | 函数的嵌套调用      | (154) |
| 5.5.2 | 函数的递归调用      | (157) |
| 5.6   | 数组作为函数参数     | (160) |
| 5.6.1 | 数组元素作函数参数    | (160) |
| 5.6.2 | 数组名作函数参数     | (160) |
| 5.6.3 | 多维数组作参数      | (166) |
| 5.7   | 变量作用域        | (168) |
| 5.7.1 | 局部变量         | (168) |
| 5.7.2 | 全局变量         | (169) |
| 5.8   | 变量存储类别与生存周期  | (172) |
| 5.8.1 | 静态存储变量       | (172) |
| 5.8.2 | 动态存储变量       | (173) |
| 5.8.3 | 全局变量的存储类别    | (174) |
| 5.8.4 | 变量的生存周期      | (176) |
| 5.9   | 内部函数和外部函数    | (176) |
| 5.9.1 | 内部函数         | (176) |
| 5.9.2 | 外部函数         | (178) |
| 5.10  | 函数的综合举例      | (179) |
|       | 习题           | (188) |

**第6章 指针** ..... (190)

|       |                  |       |
|-------|------------------|-------|
| 6.1   | 指针的概念            | (190) |
| 6.2   | 指针变量             | (191) |
| 6.2.1 | 指针变量的定义          | (191) |
| 6.2.2 | 指针变量的引用          | (192) |
| 6.2.3 | 指针变量的运算          | (192) |
| 6.2.4 | 指针变量作为函数参数       | (199) |
| 6.3   | 数组与指针            | (201) |
| 6.3.1 | 指针与数组的关系         | (201) |
| 6.3.2 | 指向数组元素的指针        | (202) |
| 6.3.3 | 指针与一维数组          | (202) |
| 6.3.4 | 指针与多维数组          | (204) |
| 6.4   | 字符串与指针           | (207) |
| 6.5   | 函数与指针            | (209) |
| 6.5.1 | 指向函数的指针          | (209) |
| 6.5.2 | 把指向函数的指针变量作为函数参数 | (211) |

|                              |              |
|------------------------------|--------------|
| 6.5.3 返回值为指针的函数 .....        | (213)        |
| 6.6 指针数组和指向指针的指针 .....       | (214)        |
| 6.6.1 指针数组的概念 .....          | (214)        |
| 6.6.2 指向指针的指针 .....          | (218)        |
| 6.7 综合举例 .....               | (219)        |
| 习题 .....                     | (227)        |
| <b>第7章 结构体与共用体 .....</b>     | <b>(229)</b> |
| 7.1 结构体的概念与定义 .....          | (229)        |
| 7.1.1 结构体的定义 .....           | (229)        |
| 7.1.2 结构体变量的定义 .....         | (231)        |
| 7.1.3 结构体变量的引用 .....         | (232)        |
| 7.1.4 结构体变量的初始化 .....        | (234)        |
| 7.2 结构体数组 .....              | (236)        |
| 7.2.1 结构体数组的定义 .....         | (236)        |
| 7.2.2 结构体数组的初始化与引用 .....     | (237)        |
| 7.3 结构体与指针 .....             | (241)        |
| 7.3.1 结构体变量与指针 .....         | (241)        |
| 7.3.2 结构体数组与指针 .....         | (244)        |
| 7.4 结构体作为函数参数 .....          | (247)        |
| 7.4.1 结构体变量作为函数参数 .....      | (247)        |
| 7.4.2 指向结构体变量的指针作为函数参数 ..... | (248)        |
| 7.5 动态数据结构——链表 .....         | (250)        |
| 7.5.1 链表的建立 .....            | (250)        |
| 7.5.2 链表的遍历 .....            | (254)        |
| 7.5.3 链表的插入与删除 .....         | (255)        |
| 7.6 共用体 .....                | (260)        |
| 7.6.1 共用体变量的定义 .....         | (260)        |
| 7.6.2 共用体变量的引用 .....         | (262)        |
| 7.7 位段 .....                 | (266)        |
| 7.8 用 typedef 定义类型 .....     | (268)        |
| 7.9 综合应用举例 .....             | (270)        |
| 习题 .....                     | (275)        |
| <b>第8章 文件 .....</b>          | <b>(277)</b> |
| 8.1 文件的概念与定义 .....           | (277)        |
| 8.2 文件类型指针 .....             | (279)        |
| 8.3 文件的打开与关闭 .....           | (280)        |
| 8.3.1 文件的打开(fopen 函数) .....  | (280)        |

|  |       |
|--|-------|
| 8.3.2 文件的关闭(fclose 函数) .....                     | (281) |
| 8.4 文件的读写 .....                                  | (282) |
| 8.4.1 fputc 函数和 fgetc 函数(putc 函数和 getc 函数) ..... | (282) |
| 8.4.2 fread()函数和 fwrite()函数 .....                | (285) |
| 8.4.3 fprintf 函数和 fscanf 函数 .....                | (289) |
| 8.4.4 其他读写函数 .....                               | (291) |
| 8.5 文件的定位 .....                                  | (293) |
| 8.5.1 rewind()函数 .....                           | (293) |
| 8.5.2 fseek()函数和随机读写 .....                       | (293) |
| 8.5.3 ftell()函数 .....                            | (295) |
| 8.6 综合应用举例 .....                                 | (295) |
| 习题 .....   | (297) |
| <b>第 9 章 编译预处理</b> .....                         | (298) |
| 9.1 宏定义 .....                                    | (298) |
| 9.2 “文件包含”处理 .....                               | (304) |
| 9.3 条件编译 .....                                   | (306) |
| 习题 .....   | (309) |
| <b>附录</b> .....                                  | (311) |
| <b>参考文献</b> .....                                | (334) |

# 第 1 章 程序设计基础

本章重点介绍程序设计的基本理论、基本知识和基本方法,为读者今后更好地从事程序设计和软件开发打下良好的基础。首先介绍计算机的基本组成,其次对程序设计和程序设计语言的基本概念及发展做以概述,着重介绍结构化程序设计和面向对象程序设计方法。

## 1.1 计算机基础

计算机被人类称为是 20 世纪最伟大的发明,问世几十年来,经历了电子管计算机,晶体管计算机,集成电路(IC)计算机,大规模集成电路(LSI)计算机,“智能”化计算机 5 个发展阶段。计算机的应用很广泛,涉及到国民经济、社会生活的各个领域和各个行业,应用于科学与工程计算、数据处理与信息管理等、工业生产自动化、计算机辅助系统、人工智能、图像处理及个人服务等各个方面。

计算机技术与通信技术的结合,出现了计算机网络通信(network communication),尤其是 Internet 的快速发展,使得世界各地的人们可以互相交流,包括文字和视频的交流,缩短了彼此空间上的距离,使我们足不出户就可以了解各国的国情。同时,随着 Internet 广泛应用,远程教学、远程医疗和电子商务的发展,使我们的生活方式和生活环境发生很大的变化,远程教学和远程医疗使得边缘地区的人们也可以享受先进科学带来的好处,而电子商务的蓬勃发展,使我们可以在网上购得自己所需要的一切东西。当今社会已步入 21 世纪的信息时代,了解计算机,学会使用计算机是时代对我们的要求。

计算机系统是一个很复杂的系统,主要由两大部分组成:硬件系统和软件系统。硬件系统是计算机的物质基础,软件系统是计算机得以运行的保障。

### 1.1.1 计算机硬件系统

计算机硬件系统是由各种物理部件组成的,是指计算机的硬件实体。直观地看,计算机硬件系统就是一大堆物理设备。一台计算机从硬件系统看主要由四个部件组成:中央处理器、存储器、输入设备和输出设备,如图 1.1 所示(图中实线表示数据线,虚线表示控制线)。

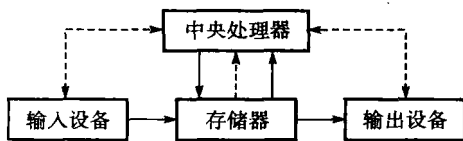


图 1.1 计算机的基本部件

#### 1. 存储器

存储器,是计算机用来存放程序和数据以及运行时数据的记忆设备。根据存储器和中央处理器的关系,存储器可分为主存储器(简称主存,又称内存)和外存储器(简称外存,又称辅存)。

主存储器通常设置主机内部,它的基本功能是按指定的位置(地址)写入或读出(也称访

问 access)信息。主存储器可以看成是由一系列存储单元组成的,每个存储单元都有一个特定的编号,即地址,地址指出了该单元在主存储器中的相对位置,便于中央处理器进行访问。每个存储单元可以存一个数据代码,该代码可以是指令,也可以是数据。计算机计算之前,程序和数据通过输入设备送入主存储器,计算开始后,主存储器不仅要为其它部件提供必需的信息,也要保存运算中间结果及最后结果。总之,它要和各个部件打交道,进行数据传送。主存储器是用半导体器件构造成的,如常说的 RAM,其存取速度比较快,但由于价格上的原因,容量较小。

外存储器设置在主机外部,用来存放相对不经常使用的程序和数据。外存储器的容量一般较大,而且可以移动,如移动硬盘等,采用 usb 接口,更加方便计算机之间进行信息交换,而价格较低,存取速度较慢,但现在速度提高比较快,可达 10 000 r/s 以上,常用的外存有磁盘、磁带和光盘。磁盘又分为软盘和硬盘。

## 2. 中央处理器

中央处理器简称 CPU(central processing unit),它是整个计算机的核心,计算机发生的所有动作都是受 CPU 控制的。CPU 主要包括运算器、控制器和寄存器三个部分。

运算器也称为算术逻辑部件 ALU(arithmetic logic unit),主要完成数据的算术运算(如加、减、乘、除)和逻辑运算(如与、或、非运算)。控制器负责从内存中读取各种指令,并对指令进行分析,根据指令的具体要求向计算机各部件发出相应的控制信息,协调它们的工作,从而完成对整个计算机系统的控制,因此控制器是计算机的指挥控制中心。CPU 内部的寄存器主要用来存放经常使用的数据。

## 3. 输入设备

输入设备的主要功能是把要输入的程序和数据信息通过输入接口顺序的送往计算机存储器中。常用的输入设备有键盘、鼠标、扫描仪、手写笔、摄像头等。

## 4. 输出设备

输入/输出设备,又称为 I/O 设备。输出设备的主要功能是将计算机操作的结果,转换为能识别和接收的信息形式,通过输出接口输送出来。常用的输出设备有显示器、打印机、绘图仪等。磁盘既是输入设备,又是输出设备。

### 1.1.2 计算机软件系统

仅有硬件系统的计算机(常称为裸机)是无法工作的,必须为它编制由一条条指令组成的各种程序才能正常工作。计算机软件系统就是为了运行、管理和维护计算机而编制的各种程序的总和,是指使用和发挥计算机效能的各种程序的总称。软件系统是计算机的灵魂,主要分为系统软件和应用软件。

#### 1. 系统软件

系统软件是主要用来对计算机系统实际运行进行控制,管理和服务的,是为了发挥计算机各部件的功能,方便用户使用而编制的,能实现系统功能的软件,主要包括操作系统 OS(operating system)、数据库管理系统、各种语言的编译系统和解释系统以及各种工具软件等。

操作系统是对裸机的第一层扩充,是管理和控制计算机资源,如 CPU、I/O 设备及存储器等,使应用程序得以自动执行的程序,最常用的操作系统有 DOS、Windows、Unix、Linus 等。

数据库管理系统是指用来帮助用户在计算机上建立、使用和管理数据的系统软件。各种语言的编译系统和解释系统是程序设计语言的翻译系统。工具软件有时又称服务软件,是开发和研制各种软件的工具。常见的工具软件有诊断程序(如杀毒软件)、调试程序、编辑程序等。

## 2. 应用软件

应用软件是用户利用计算机及其提供的系统软件,为解决实际问题而编制的计算机程序。由于计算机的日益普及,各行各业、各个领域的应用软件越来越多,常用的应用软件有以下几种:

- (1) 信息管理软件,如 MIS(管理信息系统);
- (2) 办公自动化系统,如 OFFICE、WPS;
- (3) 辅助设计软件以及辅助教学软件,如 AUTOCAD;
- (4) 图像处理软件,如 PHOTOSHOP、COREDRAW;
- (5) 数值处理软件,如 MATHLAB;
- (6) 网络应用及影音处理软件,如 IE、RealPlayer 等。

用户可以根据需要解决的各种实际问题,选择不同的应用软件。

### 1.1.3 计算机的发展

世界上第一台计算机是 1946 年美国研制成功的全自动电子数字计算机 ENIAC。这台计算机共用了 18 000 多个电子管,占地 170 m<sup>2</sup>,总重量为 30 t,耗电 140 kw/h,每秒能作 5000 次加减运算。这台计算机虽然有许多明显的不足之处,它的功能还不及现在的一台普通微型计算机,但它的诞生宣布了电子计算机时代的到来,其重要意义在于它奠定了计算机发展的基础,开辟了一个计算机科学技术的新纪元。

在短短的半个多世纪中,计算机的发展突飞猛进,经历了电子管、晶体管、集成电路和超大规模集成电路四个阶段,计算机的体积越来越小,功能越来越强,价格越来越低,应用越来越广泛。按照电子元器件的发展阶段,可将计算机分为四代。

#### 1. 第一代计算机

第一代计算机是以第一台计算机 ENIAC 问世开始到 20 世纪 40 年代末。这一时期的计算机的主要特征是使用电子管作为基本器件;软件还处于初始阶段,使用机器语言与符号语言编制程序。

第一代计算机是计算机发展的初级阶段,其体积比较大,运算速度也比较低,存储容量不大,使得所编制的程序很复杂。这一代计算机主要用于科学计算。

#### 2. 第二代计算机

第二代计算机是从 20 世纪 50 年代末到 60 年代初,其中 1958 年与 1959 年是这一代计算机的鼎盛时期。这一时期的计算机的主要特征是使用晶体管作为基本器件,在软件方面开始使用计算机高级语言,为更多的人学习和使用计算机铺平了道路。

这一代计算机的体积大大减小,具有重量轻、寿命长、耗电少、运算速度快、存储容量比较大等优点。因此,这一代计算机不仅用于科学计算,还用于数据处理和事务处理,并逐渐用于工业控制。

### 3. 第三代计算机

第三代计算机是从 20 世纪 60 年代中期到 70 年代初期。这一时期的计算机的主要特征是使用中、小规模集成电路作为基本器件,由于操作系统的出现,使计算机的功能越来越强,应用范围越来越广。

使用中、小规模集成电路制成的计算机,其体积得到了进一步的减小,功能得到了进一步的增强,可靠性和运算速度等指标也得到了进一步的提高,并且为计算机的小型化、微型化提供了良好的条件。这一时期中,计算机不仅用于科学计算,还用于文字处理、企业管理、自动控制等领域,出现了计算机技术与通信技术相结合的信息管理系统,可用于生产管理、交通管理、情报检索等领域。另外,微型计算机得到了飞速的发展,对计算机的普及起到了决定性的作用。

### 4. 第四代计算机

第四代计算机是指用大规模与超大规模集成电路作为电子器件制成的计算机。这一代计算机在各种性能上都得到了大幅度的提高,对应的软件也越来越丰富,其应用已经涉及到国民经济的各个领域,已经在办公自动化、数据库管理、图像识别、语音识别、专家系统等众多领域中大显身手,并且已进入了家庭。

#### 1.1.4 计算机的发展方向

计算机的广泛应用有力地推动了国民经济的发展和科学技术的进步,同时也对计算机技术提出了更高的要求,从而促进了计算机的进一步发展。以超大规模集成电路为基础,未来的计算机将向巨型化、微型化、网络化与智能化的方向发展。

##### 1. 巨型化

巨型化并不是指计算机的体积大,而是指计算机的运算速度更高、存储量更大、功能更强。为了满足如天文、气象、宇航、核反应等科学技术发展的需要,也为了满足计算机能模拟人脑学习、推理等功能所必需的大量信息记忆的需要,必须发展超大型的计算机。目前正在研制的巨型计算机,其运算速度可达每秒千亿次,内存容量可达几百 MB,而外存的容量将更大,这样的巨型计算机,其信息存储的能力可超过一般大型图书馆所需要的信息存储量。

##### 2. 微型化

超大规模集成电路的出现,为计算机的微型化创造了有利条件。目前,微型计算机已进入仪器、仪表、家用电器等小型仪器设备中,同时也作为工业控制过程的“心脏”,使仪器设备实现“智能化”,从而使整个设备的体积大大缩小,重量大大减轻。自 20 世纪 70 年代微型计算机问世以来,大量小巧、灵便、物美价廉的个人计算机为计算机的应用普及做出了巨大的贡献。随着微电子技术的进一步发展,个人计算机将发展得更加迅速,其中笔记本型、掌上型、PDA 等微型计算机以更优的性能和更低的价格日益受到人们的欢迎。

##### 3. 网络化

随着计算机应用的深入,特别是家用计算机越来越普及,一方面希望众多用户能共享信息资源,另一方面也希望计算机之间能互相传递信息进行通信。个人计算机的硬件和软件配置一般都比较低,其功能也有限,因此,要求大型与巨型计算机的硬件和软件资源以及它们所管

理的信息资源应该为众多的微型计算机所共享,以便充分利用这些资源。这些原因,促使计算机向网络化发展,将分散的计算机连接成网,组成计算机网络。在计算机网络中,通过网络服务器,一台台计算机就像人体的一个个神经单元被联系起来,从而组成信息社会的一个重要的神经系统。

计算机网络是现代通信技术与计算机技术相结合的产物。所谓计算机网络,就是把分布在不同地理区域的计算机与专门的外部设备用通信线路互连成一个规模大、功能强的网络系统,从而使众多的计算机可以方便地互相传递信息,共享硬件、软件、数据信息等资源。计算机网络技术是在20世纪60年代末、70年代初开始发展起来的,由于它符合社会发展的趋势,因此其发展的速度很快。目前,已经出现了许多局部网络产品,应用已经比较普遍,尤其是在现代企业的管理中发挥着越来越重要的作用。实际上,像银行系统、商业系统、交通运输系统等部门,要真正实现自动化,具有快速反应能力,都离不开信息传输,离不开计算机网络。

社会及科学技术的发展,对计算机网络的发展提出了更高的要求,同时也为其发展提供了更加有利的条件。计算机网络与通信网的结合,可以使众多的个人计算机不仅能够同时处理文字、数据、图像、声音等信息,而且可以使这些信息四通八达,及时地与全国乃至全世界的信息进行交换。同时,随着互联网的发展,出现了网格(grid)这一新兴技术,把整个互联网整合成一台超级计算机,从而实现计算资源、存储资源、通信资源、软件资源、信息资源、知识资源的全面共享,用户可以在世界任何角落访问和使用这些资源。

#### 4. 智能化

最初,计算机主要用于计算。但是,现代计算机的作用早已突破了“计算”这一初级含义。

计算机人工智能的研究是建立在现代科学基础之上的。计算机智能化程度越高,就越能代替人的作用。因此,智能化是计算机发展的一个重要方向。现在正在研制的新一代计算机,要求它能模拟人的感觉行为和思维过程的机理,使计算机不仅能够根据人的指挥进行工作,而且还会“看”、“听”、“说”、“想”、“做”,具有逻辑推理、学习与证明的能力。这样的新一代计算机是智能型的,甚至是超智能型的,它具有主动性,具有人的部分功能,不仅可以代替人进行一般工作,还能代替人的部分脑力劳动。

现在,世界上许多国家都在积极开展智能型计算机的研制开发工作,这是人类对计算机技术的一种挑战,也是对其它有关领域和学科发起的挑战,它必将促进其它众多学科的进一步发展。

## 1.2 程序设计基础

当我们需要利用计算机来解决一个实际问题时,必须依靠一定的工具(人机界面)编写一个指令(语句)清单,一旦指令(语句)清单提交给计算机,计算机就可以执行这些指令(语句),产生结果。程序就是计算机可以执行的指令序列或语句序列。而设计、编制、调试程序的过程就称之为程序设计。编写程序所使用的语言即为程序设计语言,它为程序设计提供了一定的语法和语义,所编写出的程序必须严格遵守它的语法规则,这样编写出的程序才能被计算机所接受和运行,才能产生预期结果。



### 1.2.1 程序及算法

一个程序应包括以下两方面内容:

(1)对数据的描述。在程序中要指定数据的类型和数据的组织形式,即数据结构(data structure)。

(2)对数据操作的描述。即操作步骤,也就是算法(algorithm)。

数据是操作的对象,操作的目的是对数据进行加工处理,以得到期望的结果。打个比方,厨师做菜时,需要有菜谱。菜谱上一般应包括:①配料,指出应使用哪些原料;②操作步骤,指出如何使用这些原料按规定的步骤加工成所需的菜肴。面对同一些原料可以加工出不同风味的菜肴。作为程序设计人员,必须认真考虑和设计数据结构和操作步骤(即算法)。因此,著名计算机科学家沃思(Nikiklaus Wirth)提出一个公式

$$\text{数据结构} + \text{算法} = \text{程序}$$

实际上,一个程序除了以上两个主要要素之外,还应当采用结构化程序设计方法进行程序设计,并且用某一种计算机语言表示。因此,还可以这样表示

$$\text{程序} = \text{算法} + \text{数据结构} + \text{程序设计方法} + \text{语言工具和环境}$$

也就是说,以上四个方面是一个程序设计人员所应具备的知识。在设计一个程序时要综合运用这几方面的知识。在这四个方面中,算法是灵魂,数据结构是加工对象,语言是工具,编程需要采用合适的方法。算法是解决“做什么”和“怎么做”的问题。程序中的操作语句,实际上就是算法的体现。显然,不了解算法就谈不上程序设计。下面用例子来引入算法的概念。

**例 1.1** 给定两个正整数  $p$  和  $q$ ,求  $p$  和  $q$  的最大公约数  $g$ 。

如何求得两个正整数的最大公约数?对于这样一个数学问题,数学家欧几里得(Euclid)给出了一个解决方案,这个方案由三个步骤完成,如下所示。

求解最大公约数的欧几里德算法

步骤 1:如果  $p < q$ ,则交换  $p$  和  $q$

步骤 2:令  $r$  是  $p/q$  的余数

步骤 3:如果  $r = 0$ ,则令  $g = q$ ,结束算法, $g$  即为求得的最大公约数;

否则令  $p = q, q = r$ ,转向步骤 2。

按照以上的步骤,就可以计算出任意两个正整数的最大公约数。例如求 24 和 16 的最大公约数,即  $p = 24, q = 16$ 。从步骤 1 开始执行,显然  $p > q$ ,不满足条件  $p < q$ ,因此不用交换  $p$  和  $q$ ;执行步骤 2,  $24/16$  的余数是 8,即  $r = 8$ ;执行步骤 3,  $r \neq 0$ ,则  $p = 16, q = 8$ ,转向步骤 2,  $16/8$  的余数是 0,即  $r = 0$ ;接着执行步骤 3,因为  $r = 0$ ,则  $g = 8$ ,算法结束。最后算得的  $g = 8$  即为 24 和 16 的最大公约数。计算工具可以是笔和纸,也可以是计算机。

我们把这种将问题归结为有规律的操作步骤,并且用有限多个步骤来表示的具体过程就称之为算法。对同一个问题,可以有不同的解题方法和步骤。

**例 1.2** 求  $1+2+3+\dots+100$

算法 1

步骤 1:  $1+2 = 3$

步骤 2:  $3+3 = 6$

步骤 3:  $6+4 = 10$