

高等院校精品课程系列教材

计算机程序设计

(C语言版)

贾伯琪 顾为兵 苏仕华 张四海 何克东◎编著



*Fundamentals of Computer
Programming in C*



机械工业出版社
China Machine Press

高等院校精品课程系列教材

计算机程序设计

(C语言版)

贾伯琪 顾为兵 苏仕华 张四海 何克东 编著



*Fundamentals of Computer
Programming in C*



机械工业出版社
China Machine Press

本书选用 Linux 和 Microsoft Visual C++ 6.0 作为教学环境,以 C 程序设计语言为载体,展示了在程序设计过程中如何分析问题、组织数据并解决问题,阐述了在计算应用过程中将算法和编码相联系的具体程序设计过程,并向读者介绍计算机结构化程序设计的基本概念、技术和方法。本书内容全面,从逻辑上可分三大部分:第一部分由第 1 章和第 2 章组成,主要包括计算机系统组成及基本工作原理、微型机系统结构、信息的表示及存储形式、程序与算法的基本概念、C 语言的程序结构;第二部分由第 3~12 章组成,主要介绍数据类型、运算符和表达式、控制结构、数组、函数、预处理、指针、结构体与共用体、位运算、流与文件操作、面向对象程序设计简介;第三部分由 4 个附录组成,主要包括 C 语言常用的库函数、ASCII 码表、C 语言的开发环境及上机过程介绍等知识。

本书可作为高等院校计算机及相关专业本科生的 C 语言教材,也适合教师和专业人士参考。

封底无防伪标均为盗版

版权所有,侵权必究

本书法律顾问 北京市展达律师事务所

图书在版编目(CIP)数据

计算机程序设计:C语言版/贾伯琪等编著. —北京:机械工业出版社,2011.8
(高等院校精品课程系列教材)

ISBN 978-7-111-35351-5

I. 计… II. 贾… III. C语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2011)第 139613 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:王璐 李荣

计算机程序设计是计算机教学系列课程中的重要组成部分。本课程的主要任务是以培养学生计算机应用能力和计算思维能力为目标,引导学生进入计算机程序设计的广阔天地。当今社会已进入信息化时代,善于运用计算机技术和手段进行学习、工作、解决专业问题已经是高级人才必备的素质。计算机程序设计教学不仅是大学通识教育的一个重要组成部分,更是培养大学生用计算思维方式解决专业问题、成为复合型创新人才的基础性教育的关键环节。

本书从结构化程序设计方法出发,以C程序设计语言为载体,通过对典型问题的分析、算法描述和相应的C语言代码实现,展示了在程序设计过程中如何分析问题、组织数据并解决问题,阐述了在计算应用过程中将算法和编码相联系的具体程序设计过程,进而向读者介绍计算机结构化程序设计的基本概念、技术和方法,以启发读者使用计算机解决问题所需的思路和算法思想,培养读者的逻辑思维能力、抽象能力和计算机程序设计能力。

本书通过部分实例与相关专业应用的有机结合,揭示了程序设计与后续课程以及与专业应用之间的密切联系,帮助读者学以致用,明确学习目标,并激发学习动力。因此,本书讲解的计算机程序设计不仅为解决专业领域的问题提供了有效的方法和手段,还提供了一种独特的处理问题的思维方式。

本书采用基础优先的策略,全书内容从逻辑上可分三大部分:第一部分由第1章和第2章组成,涵盖了学习程序设计的预备知识,主要包括计算机系统组成及基本工作原理、微型机系统结构、信息的表示及存储形式、程序与算法的基本概念、C语言的程序结构;第二部分由第3~12章组成,主要介绍数据类型、运算符和表达式、控制结构、数组、函数、预处理、指针、结构体与共用体、位运算、流与文件操作、面向对象程序设计简介;第三部分涵盖了4个附录,主要包括C语言常用的库函数、ASCII码表、C语言的开发环境及上机过程介绍等知识。全书以计算机程序设计的思想贯穿始终,并以计算思维能力的培养为目标,引导读者从宏观上把握程序设计的理念。

从计算机学科的整体培养目标来看,程序设计不仅要求学生学习语法,更要求学生学会解决实际问题,本书最重要的是教会初学者在解决问题时使用多种层次的抽象,能从高、中、低多个层面看待问题。程序设计课程与计算机学科的其他课程有很大区别,在程序设计课程中,学生要从案例中学习、从实践中学习、从错误中学习,需要花费大量时间编写、调试并修改程序。从我们计算机程序设计课程组诸多教师的经验看来,对于初学编程的人,不仅必须掌握数据与数据之间的关系,以及数据在计算机内的表示、数据的操

作、基础逻辑和循环的基本编程技术，还需要强化上机实践。只有通过实践才能切实感受和领悟用计算机进行问题求解的思维模式和基本方法。

对于刚接触程序设计的新生来说，在掌握了编写程序的基本技能后，就应该重点理解和运用结构化程序设计的思想和方法。在程序设计课程教学过程中，很多人容易陷入一个误区，即只注重语言本身的学习，以为学会了语法、规则、指令就等于掌握了程序设计，却忽略了计算思维的培养。然而，经验表明该课程的教学应该突出使用编程解决特定问题的方式，即程序设计方法，而不是语言本身。

本书内容全面，在知识的讲解上又不失趣味性，作为教材，既适合作为初学者的入门读物，符合初学者的学习特点和过程，也适合教师和专业人士参考。如果读者耐心地读完本书，会发现本书在内容上具有较大的跨度和深度。

本书选用 Linux 和 Microsoft Visual C++ 6.0 作为教学环境，为了方便读者学习，在本书的附录中提供了 ASCII 码对照表、常用标准库函数、C 程序的上机过程与 UNIX 系统使用，以及 VC++ 6.0 集成环境开发 C 程序的基本方法。

本书由贾伯琪、顾为兵、苏仕华、张四海和何克东编著，具体分工如下：顾为兵完成第 3 章和第 7 章，苏仕华完成第 4 章、第 9 章和第 12 章，张四海编写第 6 章和第 10 章并提供部分与专业结合的实例，何克东编写第 1 章，贾伯琪完成其他章节并负责对全书内容进行调整、修改和统一定稿。

感谢计算机程序设计课程组的帅建梅、尹东、王上飞、谭立湘、刘勇、白雪飞、郑重、吴文涛、秦琳琳、杨坚等老师的大力支持和帮助。

感谢中科大近代物理系的邓友金教授和刘乃乐副教授，中科大生命科学学院的梁治博士后，中科大信息科学技术学院的周武旸教授，赵明老师，他们为本书的实例研究提供了有趣的问题素材，与他们的讨论对该部分内容的完成具有重要影响。

鉴于编者水平和成书时间有限，书中错误和不妥之处在所难免，敬请同行教师和读者批评指正。读者可以发 E-mail 到 bqjia@ustc.edu.cn 与作者沟通。

编 者

2011 年 7 月于中国科学技术大学

教学建议

教学内容	学 习 要 点 及 教 学 要 求	教学安排		
		主讲		实验
		56	36	40
第 1 章 计算机与程序 设计基础	(1) 理解计算机的基本组成及各基本部件的功能 (2) 理解计算机的基本工作原理(存储程序原理) (3) 理解存储器的基本结构、存储单元、地址及按地址访问数据、指令和程序的概念 (4) 理解计算机中信息的表示方法 (5) 了解计算机的数字系统、计数法及进制的概念 (6) 掌握十进制、二进制、八进制、十六进制数据之间的转换 (7) 理解正负数、实数的浮点表示及值域和精度的概念 (8) 理解计算机中非数值信息的表示方法(ASCII 码、汉字编码等)	4	4	—
第 2 章 C 程序结构和 设计方法	(1) 了解常用程序设计语言的特点 (2) 理解程序的组成元素、基本语法、基本结构和程序的准确书写格式 (3) 了解程序的解释执行和编译执行 (4) 理解程序设计的基本过程	2	2	4
第 3 章 数据类型、运算符 和表达式	(1) 理解基本数据类型的定义和使用方法 (2) 理解变量、常量的概念和语法规则 (3) 理解基本运算符与表达式, 以及运算符的优先级和结合性 (4) 掌握基本输入与输出方法	6	4	4
第 4 章 控制结构与 基本算法	(1) 理解控制语句的作用、分类和三种主要的程序控制结构 (2) 理解控制语句的嵌套结构和复杂控制结构的程序设计方法 (3) 理解算法的概念和算法的描述方法 (4) 理解算法(如累加、递推、数制转换等)的程序控制结构	8	4	4
第 5 章 数 组	(1) 理解数组的定义、初始化和引用方法 (2) 理解与数组有关的常用算法(如冒泡排序和选择排序、二分查找算法等) (3) 了解数组在数值计算、数据统计、数据检索方面的应用 (4) 理解字符数组的概念以及字符串处理函数的编写与使用方法 (5) 理解数组作为过程/函数参数的概念和调用函数处理数组的方法	6	5	4
第 6 章 函 数	(1) 理解过程/函数的定义方法 (2) 理解过程/函数调用和过程/函数参数(值传递与地址传递)的概念和传递机制 (3) 理解过程/函数的嵌套调用和递归调用 (4) 理解变量的生存周期、作用域的概念以及局部变量和全局变量的使用方法	6	4	4

(续)

教学内容	学 习 要 点 及 教 学 要 求	教学安排		
		主讲		实验
		56	36	40
第 6 章 函 数	(5) 理解变量的静态存储和动态存储方式 (6) 理解模块化程序设计的基本思想和利用过程/函数实现模块化设计的方法	6	4	4
第 7 章 预处理命令	(1) 理解预处理指令的种类和作用 (2) 理解包含指令和宏定义指令, 了解条件编译指令	2	1	2
第 8 章 指 针	(1) 理解指针的基本概念和指针变量的定义与使用方法 (2) 理解指针和数组的关系、指针数组的使用方法和指针在数组操作中的应用方法 (3) 理解字符串的概念以及指针在字符串操作中的使用方法 (4) 理解指针作函数参数的用法	7	5	4
第 9 章 结构体与共用体	(1) 理解结构体类型的概念以及结构体变量和结构体数组的使用方法 (2) 掌握使用结构体数组实现线性表的方法 (3) 掌握使用链表(建立、遍历、插入、删除等操作)实现线性表的方法 (4) 理解共用体类型的概念, 了解结构体和共用体嵌套结构体的使用方法 (5) 理解枚举类型的概念和枚举变量的使用方法	7	4	4
第 10 章 位 运 算	(1) 理解基本位运算符 (2) 掌握位运算的方法	2	1	2
第 11 章 流与文件操作	(1) 理解文件指针 (2) 掌握文件的打开与关闭、文件读写、输入/输出格式控制、输入/输出函数的使用方法	4	2	4
*第 12 章 C++语言概述 (选学)	了解面向对象的有关概念及面向对象程序设计思想、方法和重要性等	2	—	上机 考核 4

说明:

1. 中国科学技术大学计算机程序设计教学课程组多年来以培养学生计算机应用能力和计算思维能力为目标, 在计算机基础教学能力结构、知识体系、实验体系、课程设置和教学方法等方面做了一些工作。
2009 年在校本科教学工作会议的计算机基础课程教学改革专题研讨会上提出了 C 语言程序设计基础课的改革设想。根据我国计算机普及教育的状况和课程组教师的教学体验, 分析本课程教学改革现状、动态和进展, 提出了从学科方法论上打基础, 培养计算思维能力和分析问题的能力等教学理念, 并广泛征求各院系教学主管部门的意见, 得到学校和各院系的支持。由此课程改革取得较大进展, 有了统一明确的教改方向和具体改革方案, 对原有的计算机基础课程设置进行了调整。原“计算机文化基础”课不再作为通识课, 原“C 语言程序设计”课程改名为“计算机程序设计”, 并分 A 型(60/40)和 B 型(60/60)分层次教学。继续实施本课程的全校统一教学大纲、统一教材、统一考试和阅卷。
教学实践表明成效显著。自 2005 年我校教务处对毕业生“您感觉受益最深的课程”调查以来, 从公布的结果看, “C 语言程序设计”课程连续六年排在前 10 名, 其中 2009 届排名第二, 2010 届排名第三。本科教学检查及课堂教学学生问卷结果表明, 学生对本课程的课堂教学满意度不断提升, 2007~2008 学年、2008~2009 学年课堂满意数占课堂总数的 50%以上, 2009~2010 学年课堂满意数占课堂总数的 66.67%, 2010~2011 学年(课程调整后)课堂满意数占课堂总数的 95%。
2. 本教材主要是为我校新生在第一学期开设的“计算机程序设计”通识课而编写的, 建议课堂授课 40~60 学时, 实验课 40 学时, 不同的学校可以根据各自的教学要求和计划学时数酌情对教材内容进行取舍。
3. 在学习计算机程序设计课程之前, 若已开设先修课“计算机基础”, 则可跳过第 1 章直接讲授后续章节, 其学时可用于重点章节的讲授。

目 录

前言	
教学建议	
第1章 计算机与程序设计基础	1
1.1 引言	1
1.2 计算机系统的基本组成与工作原理	2
1.2.1 计算机系统的组成	2
1.2.2 计算机的基本工作原理	4
1.3 微型计算机的系统结构	5
1.3.1 微型计算机的主要性能指标	5
1.3.2 微型计算机的硬件组成	5
1.3.3 微型计算机的软件系统	9
1.4 计算机中信息的表示及存储形式	13
1.4.1 计算机采用二进制编码	13
1.4.2 计算机中的信息单位	14
1.4.3 计算机中的数字系统	14
1.5 计算机程序与算法	23
1.5.1 算法的概念	23
1.5.2 算法的基本结构和表示	27
1.5.3 算法设计原则	31
1.5.4 算法的基本分类	32
1.6 程序设计和程序设计语言	35
1.7 本章小结	36
1.8 习题	37
第2章 C 程序结构和设计方法	38
2.1 引言	38
2.2 C 语言概述	39
2.3 C 程序的基本结构	40
2.3.1 预处理部分	40
2.3.2 程序基本部分	40
2.4 程序设计方法	44
2.4.1 明确问题的需求	44
2.4.2 分析问题	46
2.4.3 设计解决问题的算法	46
2.4.4 测试并验证整个程序	50
2.4.5 维护并更新程序	52
2.5 实例研究	52
2.6 本章小结	58
2.7 习题	58
第3章 数据类型、运算符和表达式	59
3.1 引言	59
3.2 对象、变量和常量的概念	59
3.3 整型数据	61
3.3.1 整型数据的分类	61
3.3.2 整型数据所占内存空间的大小和取值范围	62
3.3.3 整型变量的溢出	62
3.3.4 整数常量的表示方法	63
3.4 浮点型数据	64
3.4.1 浮点数据的分类、取值范围和精度	64
3.4.2 浮点数的误差	65
3.4.3 浮点常量的表示方法	65
3.5 字符型数据	66
3.5.1 字符数据类型	66
3.5.2 字符常量	67
3.5.3 字符串常量	68
3.6 void 数据类型	69
3.7 表达式和运算符	69
3.7.1 运算符的优先级和结合方向	70
3.7.2 左值的概念	71
3.7.3 几种运算符简介	72
3.8 表达式中类型的隐式转换和显式转换	78

3.9 基本输入输出	81	5.2.3 一维数组的初始化	117
3.9.1 格式化输出库函数 printf()的使用	81	5.2.4 一维数组作为函数参数	123
3.9.2 格式化输入库函数 scanf()的使用	82	5.3 字符数组	127
3.9.3 案例研究	83	5.3.1 字符数组与字符串的关系	127
3.10 本章小结	84	5.3.2 字符数组的定义和引用	127
3.11 习题	84	5.3.3 字符数组的初始化	127
第4章 控制结构与基本算法	86	5.3.4 字符数组应用编程 (字符串处理)	130
4.1 引言	86	5.4 多维数组	134
4.2 C语言语句	88	5.4.1 二维数组的定义	134
4.2.1 表达式语句	88	5.4.2 二维数组的内部表示	135
4.2.2 函数调用语句	89	5.4.3 二维数组的引用	135
4.2.3 控制语句	89	5.4.4 二维数组的初始化	137
4.2.4 复合语句	89	5.4.5 二维数组作为函数参数	141
4.2.5 空语句	89	5.5 数组类型的应用	144
4.3 选择结构	89	5.5.1 数据统计应用	144
4.3.1 if 语句	89	5.5.2 排序算法应用	145
4.3.2 switch 语句	94	5.5.3 查找算法应用	151
4.4 循环控制结构	97	5.5.4 字符文本处理应用	154
4.4.1 while 循环语句	97	5.5.5 矩阵处理应用	156
4.4.2 do ~ while 语句	99	5.5.6 递推与迭代算法应用	160
4.4.3 for 循环语句	99	5.6 本章小结	162
4.4.4 三种循环语句的比较	102	5.7 习题	163
4.4.5 循环的嵌套	102	第6章 函数	166
4.5 控制转向语句	104	6.1 引言	166
4.5.1 break 语句	104	6.2 函数定义的一般形式与 函数原型声明	167
4.5.2 continue 语句	106	6.3 函数调用	169
4.5.3 语句标号和 goto 语句	106	6.3.1 函数调用的一般形式及 函数的调用方式	170
4.6 常用基本算法的程序设计	107	6.3.2 函数调用中的数值 参数传递	171
4.6.1 递推法	107	6.3.3 函数的返回值	176
4.6.2 贪心法	109	6.3.4 嵌套调用	178
4.6.3 枚举法	110	6.4 函数的递归调用	179
4.7 本章小结	111	6.5 变量存储空间	181
4.8 习题	112	6.5.1 运行时环境简介	181
第5章 数组	114	6.5.2 局部变量和全局变量	181
5.1 引言	114	6.5.3 动态存储和静态存储	182
5.2 一维数组	115	6.6 内部函数和外部函数	184
5.2.1 一维数组的定义	115		
5.2.2 一维数组的引用	116		

6.6.1 外部函数	184	8.12 指针数组作 main 函数的 形参	237
6.6.2 内部函数	184	8.13 指针函数	239
6.7 实例研究	184	8.14 指向 void 量的指针变量	241
6.7.1 CDMA 系统中码字 生成问题	185	8.15 指向函数的指针	244
6.7.2 DNA 序列的翻译问题	186	8.15.1 指向函数的指针变量的 初始化和使用	245
6.7.3 集合元素的全排列问题	190	8.15.2 指向函数的指针变量 作为函数参数	245
6.8 本章小结	193	8.16 动态分配内存空间和 动态数组	250
6.9 习题	194	8.16.1 指针变量与动态 内存分配	250
第 7 章 预处理命令	196	8.16.2 一维动态数组的 建立和使用	251
7.1 引言	196	8.16.3 二维动态数组的 建立和使用	253
7.2 宏定义	197	8.16.4 C99 标准允许的 可变长数组	254
7.2.1 不带形参的宏定义	197	8.17 本章小结	255
7.2.2 带形参的宏定义	198	8.18 习题	256
7.3 文件包含	200	第 9 章 结构体与共用体	258
7.4 条件编译	201	9.1 引言	258
7.5 本章小结	203	9.1.1 结构类型定义	258
7.6 习题	204	9.1.2 定义结构类型时需要 注意的问题	259
第 8 章 指针	205	9.2 结构变量的定义、引用、 初始化及赋值	260
8.1 引言	205	9.2.1 结构变量的定义	260
8.2 指针的概念	205	9.2.2 结构变量的引用	260
8.3 指针变量的定义	207	9.2.3 结构变量的初始化	261
8.4 指针变量的引用	208	9.2.4 结构变量的赋值	261
8.4.1 &运算符	208	9.3 结构体数组及其应用	262
8.4.2 *运算符	209	9.3.1 定义结构数组	262
8.5 指针变量的运算	210	9.3.2 结构数组的初始化	262
8.5.1 空指针和无效指针	210	9.3.3 指向结构数组元素的指针	263
8.5.2 指针运算	211	9.3.4 结构数组的应用举例	263
8.5.3 关系比较运算	214	9.4 结构与链表	265
8.6 指向数组元素的指针	214	9.4.1 结构的嵌套	266
8.7 指针作为函数参数	220	9.4.2 链表的建立和访问	266
8.8 指向字符串的指针变量	224		
8.9 指向多维数组的指针变量	226		
8.9.1 指向数组元素的指针变量	228		
8.9.2 指向行向量的指针变量	229		
8.10 指针数组	233		
8.10.1 指针数组的定义	233		
8.10.2 指针数组的两个 典型实例	233		
8.11 指向指针的指针	235		

9.4.3 链表结点的插入和删除	269	11.4.1 文件的打开	294
9.4.4 链表综合实例——职工 信息管理系统	271	11.4.2 文件的关闭	295
9.5 共用体和枚举类型	273	11.4.3 标准流指针	295
9.5.1 共用体类型	273	11.4.4 文件指针与文件读写 位置指针的区别	296
9.5.2 枚举类型	276	11.5 文件的读写	296
9.5.3 类型定义符 typedef	278	11.5.1 格式化读写	296
9.6 本章小结	278	11.5.2 数据块（二进制形式）的 读写	298
9.7 习题	279	11.5.3 字符读写函数	300
第 10 章 位运算	281	11.5.4 字符串读写	302
10.1 引言	281	11.6 文件的定位和随机读写	305
10.2 基本位运算	281	11.7 文件的错误检测	308
10.2.1 按位与	282	11.8 本章小结	308
10.2.2 按位或	282	11.9 习题	308
10.2.3 按位异或	282	第 12 章 C++语言概述	310
10.2.4 取反	283	12.1 引言	310
10.2.5 左移	283	12.2 面向对象程序设计方法	310
10.2.6 右移	284	12.3 C 语言与 C++语言的关系	314
10.2.7 复合赋值运算符	284	12.3.1 C++对 C 语言的改进	314
10.3 位段	284	12.3.2 C++语言对面向对象 方法的支持	314
10.4 实例研究	285	12.4 本章小结	315
10.4.1 输出字符型数据的 二进制形式	285	12.5 习题	315
10.4.2 求二进制数中 1 的个数	288	附录 A 常用库函数	316
10.5 本章小结	290	附录 B ASCII 码表	320
10.6 习题	290	附录 C C 语言程序的上机过程与 UNIX 系统中的使用	324
第 11 章 流与文件操作	291	附录 D VC++ 6.0 集成开发环境简介	329
11.1 引言	291	参考文献	332
11.2 文件与流	292		
11.3 文件类型与文件指针	293		
11.4 文件的打开与关闭	293		

第 1 章

计算机与程序设计基础

【学习目标】

- 1) 理解计算机中信息的表示方法。
- 2) 理解计算机的基本组成及各基本部件的功能。
- 3) 理解软件的分类及常用软件的功能。
- 4) 理解计算机的基本工作原理（存储程序原理）。
- 5) 理解存储器的基本结构、存储单元、地址及按地址访问数据与指令的概念。
- 6) 理解计算机主要技术指标的含义。
- 7) 了解操作系统的发展、分类及操作系统的主要功能。
- 8) 理解文件、目录与路径的概念。
- 9) 理解计算机中存储信息的方法。
- 10) 理解计算机采用的二进制编码及信息单位。
- 11) 了解计算机的数字系统、计数法及进制的概念。
- 12) 掌握十进制、二进制、八进制、十六进制数据之间的转换。
- 13) 理解正负数、实数的浮点表示及值域和精度的概念。
- 14) 理解计算机中非数值信息的表示方法（ASCII 码、汉字编码、Unicode 码及多媒体信息编码）。
- 15) 理解算法的概念、掌握算法的描述。
- 16) 理解结构化程序设计思想、掌握结构化程序设计方法和规则。

1.1 引言

计算机技术是人类在 20 世纪最重要的发明之一。自 1946 年第一台计算机问世以来，计算机的应用已日益深入到我们社会的生产、生活、国防、科技、教学、商务、文化、娱乐等方方面面。大到宇宙飞船的发射、电子商务、电子政务的运行，小到电子邮件的收发、使用手机通话、通过网络聊天和玩电子游戏机，无不需要计算机技术的支持。随着技术的发展，计算机技术将会在人类活动的几乎所有领域得到应用，深刻而持久地改变我们的社会和生活。

计算机技术之所以具有如此大的影响力，一个重要的原因是，一台计算机由硬件系统和

软件系统两大部分构成。计算机的硬件,包括它的核心功能部件和外围设备,只是提供了一个具有广泛通用性的计算平台。计算机系统功能的多样化和复杂化,主要取决于它的软件系统。可以粗略地认为,硬件决定计算机系统的性能,而软件决定计算机系统的功能。硬件性能的提高在很大程度上取决于电子技术的进展,而软件功能的发展取决于人们的需求,以及软件工作者的想象力和程序设计能力。硬件是基础,软件是计算机的灵魂,没有软件的计算机什么都不能做,只有安装了软件的计算机,才能进行信息处理,成为一台真正意义的计算机。

最初的计算机主要用于科学计算。现在,计算机作为一种现代化的处理信息的工具,其用途早已超出了数值计算的范围,可以用于文字、图形/图像、动画和声音等多媒体数据的处理。这些数据的外在表现形式差别很大,但在计算机内部,则用统一的二进制数表示,并有特定的计算机程序将其解释成不同的文字、图形/图像、动画和声音等。计算机所能完成的工作以及怎样完成工作都是由人指定的,即计算机所做的任何工作都是在由人编写的程序的控制下进行的。

根据计算机的工作特点,可以把计算机描述成是一种能存储程序和数据、并能自动、精确、高速执行程序的现代化智能电子设备,是一种能对各种数字化信息进行处理的工具。

学习计算机程序设计,不仅要了解计算机是什么,计算机能够做什么、如何做,而且还要知道这个学科领域解决问题的基本方法与特点,即学习这个学科领域解决问题的基本思维方法。因此本章介绍的内容主要有:计算机系统组成及基本工作原理、二进制编码、计数法、进制概念、各种数制转换,同时简要介绍算法和结构化程序设计的基本概念,为读者提供计算机程序设计的一般性准则,以便于本课程的学习。

1.2 计算机系统的基本组成与工作原理

1.2.1 计算机系统的组成

一个完整的计算机系统 (Computer System) 由硬件 (Hardware) 系统和软件 (Software) 系统两大部件组成。

硬件系统是指组成计算机的具体的物理设备。1946 年,美籍匈牙利数学家冯·诺依曼在参加第一台计算机的研制中,提出了三个重要的设计思想:

- 1) 采用二进制编码形式表示数据和程序;
- 2) 要执行的程序和被处理的数据预先放入计算机中,计算机能够自动地从内存中取出指令执行;
- 3) 计算机由运算器、控制器、存储器、输入设备、输出设备五大基本部件组成 (如图 1-1 所示)。

冯·诺依曼的思想奠定了计算机科学发展的理论基础,并被应用于实际设计中,为计算机的发展立下了不朽的功勋。按照冯·诺依曼思想设计的计算机被称为冯·诺依曼结构计算机。60 多年来,计算机的性能、速度、应用领域等都发生了翻天覆地的变化,但计算机的基本结构没有变。“存储程序原理”至今仍然是计算机的基本工作原理。存储

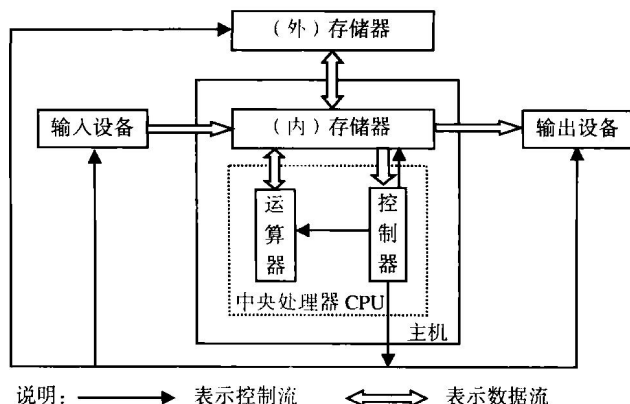


图 1-1 计算机的基本结构

程序原理的主要思想是：将程序和数据存放到计算机内部的存储器中，计算机在程序的控制下一步一步地进行处理，直到得出结果。自从计算机诞生的那一天起，这一原理就决定了人们使用计算机的主要方式——编写程序和运行程序。

冯·诺依曼计算机各大部件的名称和功能如下：

1) 运算器。算术逻辑部件，又称 ALU (Arithmetic Logic Unit)，负责完成所有算术运算、逻辑运算任务。

2) 控制器 (Control Unit)。如同人的大脑，负责指挥计算机各部件按照指令指定的功能进行各种操作，使计算机能够按照程序的安排，自动完成规定的任务。

3) 存储器 (Memory Unit)。计算机的重要部件之一，是计算机能够实现“存储程序”功能的硬件基础。程序和数据以及运算的中间结果和最终结果均以二进制形式存储在计算机的存储系统中。计算机的存储系统分为内部存储器（简称内存或主存）和外部存储器（简称外存或辅存）。内存中存放将要执行的指令和运算数据，存取速度快，但容量较小。外存容量大、成本低、存取速度慢，用于存放需要长期保存的程序和数据。当存放在外存中的程序和数据需要处理时，必须先将它们读入内存中，才能进行处理。

4) 输入设备 (Input Device)。用来接受用户输入的原始数据和程序，并将它们变为计算机能够识别的代码形式存放到存储器中。

5) 输出设备 (Output Device)。用于将存储器中经计算机处理得到的运算结果转变为人们所能接受的形式，输出到相应的介质上，供使用者观看、分析和保存。

五大部件构成了计算机的硬件部分。在微型计算机中运算器和控制器是集成在一个半导体芯片中的，通常把运算器和控制器统称为 CPU。CPU 是中央处理器 (Central Processing Unit) 的英文缩写。人们常把 CPU 和主存储器（内存）合称为主机。

计算机软件系统包括了计算机运行所需的各种程序及其有关的文档资料。计算机系统是在硬件（裸机）的基础上，通过一层层软件的运行，向用户呈现出友好的使用界面和强大的功能。根据程序的不同用途，通常把软件分为系统软件和应用软件两大类。

系统软件 (System Software) 是计算机设计制造者提供的管理计算机全部软硬件资源的软件。包括操作系统、语言处理程序、数据库管理系统、网络管理程序、工具与服务程序等。这些系统程序中，操作系统是最核心的软件，其他所有程序都要在它的支持下工作。

应用软件 (Application Software) 是专门为某一应用目的而编制的通用或专用程序，例如现在广泛使用的办公软件 Microsoft Office，还有财务、销售、管理、教学、产品设计等在各行各业运用的一些专用软件。应用软件必须在系统软件的支持下才能工作。

需要指出的是，计算机的硬件系统和软件系统之间是相辅相成、缺一不可的。没有任何软件的计算机称为裸机，裸机本身是不能完成任何功能的，只有配备一定的软件，才能发挥其功用。图 1-2 概括了计算机的组成，图 1-3 反映了用户和软件、硬件之间的关系。

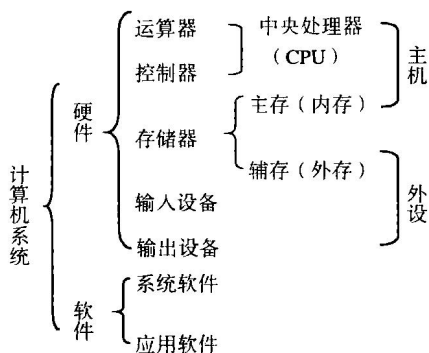


图 1-2 计算机系统的组成

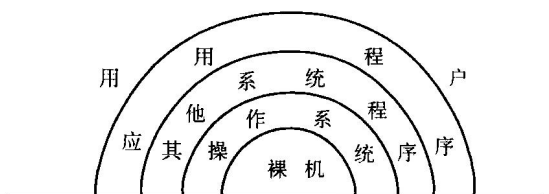


图 1-3 用户和软件、硬件之间的关系

1.2.2 计算机的基本工作原理

从图 1-1 可见, 计算机中有两股信息流在流动, 一股是数据流, 它包括参加运算的原始数据、运算过程中的中间结果和最终结果, 还包括程序中的指令; 另一股是控制流, 是由控制器发往各部件的命令。在计算机内部, 这些信息都用以“0”和“1”组合而成的二进制代码来表示, 因为计算机能理解、识别的只能是二进制信息。我们把程序和数据同时存储在存储器中, 在发出命令以后, 计算机就能够自动完成运算, 就是因为存储器中存储了程序。计算机是在程序的控制之下自动完成相应操作的, 这就是所谓的“存储程序原理”, 又称冯·诺依曼原理。为深刻理解这一原理, 下面介绍一下指令和程序的概念。

1) 指令 (instruction)。指令由一串二进制代码组成, 是计算机完成一个基本操作的命令, 如: 传送、加法、移位、停机等都是一个基本操作。指令应能由控制器中的指令译码器所识别, 因此, 不同类型的计算机由于硬件的差异, 指令代码也不同。一条指令实际上包括两种信息即操作码和操作数。操作码用来表示该指令所要完成的操作 (如加、减、乘、除、数据传送等), 其长度取决于指令系统中的指令条数。操作数用来描述该指令的操作对象, 它或者直接给出操作数, 或者指出操作数的存储器地址或寄存器地址 (即寄存器名)。指令的一般格式如图 1-4 所示。

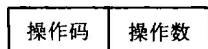


图 1-4 指令格式

2) 指令系统 (Instruction Set)。某一台计算机所能识别的所有基本指令的集合称为指令系统。指令系统的丰富与否, 决定了计算机对数据的运算和处理能力的强弱。

3) 程序 (program)。程序就是为完成预定任务用计算机语言编写的一组指令序列。计算机按照程序规定的流程依次执行指令, 最终完成程序所描述的任务。

用户在写好程序后, 要通过输入设备输入到存储器中。计算机执行程序依规定的顺序执行一条条指令, 每一条指令的执行分为两个阶段: 一是从主存储器将指令码取到 CPU 中; 二是 CPU 对取入的指令进行分析译码, 判断该指令要做什么, 然后向各部件发出完成该操作的控制信号, 使各部件产生相应的动作, 这样就完成了一条指令的执行过程, 如图 1-5 所示。



图 1-5 程序的执行过程

4) 程序设计语言 (programming language)。又称编程语言。用户通常用某种编程语言编写程序, 输入计算机, 然后由翻译软件将其转换成计算机能够理解的语言 (指令), 在计算机上运行。程序设计语言分为低级语言 (low-level language) 和高级语言 (high-level language)。低级语言又称为面向机器的语言, 如机器语言 (machine language)、汇编语言 (assembly language)。高级语言有面向过程的语言 (procedure-oriented language) 和面向对象的语言 (object-oriented language), 下面是一些常用的高级语言。

- Fortran 语言 1954 年推出, 适用于科学和工程计算。
- BASIC 语言 初学者语言, 1964 年推出; 1991 年微软推出可视化的、基于对象的 Visual Basic 开发环境, 发展到现在的 VB.NET 开发环境。VB.NET 是完全面向对象的、功能也更强大。
- Pascal 语言 结构化程序设计语言, 1968 年推出, 适用于教学、科学计算、数据处理和系统软件等开发, 目前已被 C 语言取代。
- C/C++ 语言 1972 年推出 C 语言, 1983 年语言的设计者加入面向对象概念, 改名为 C++。C/C++ 语言简练、功能强、适用面广。
- Java 语言 一种新型的跨平台的面向对象程序设计语言, 1995 年推出, 适用于网络应用。程序设计语言的发展经历了五代——机器语言、汇编语言、高级语言、非过程化语言和

智能语言。

1.3 微型计算机的系统结构

本节介绍微型计算机系统，主要介绍：硬件方面的 CPU、内存、接口和总线等概念；软件方面介绍几种常用的操作系统、操作系统功能、文件、文件夹（目录）、逻辑盘、路径的概念。

微型计算机可以分成微处理器（Micro Processor）、微型计算机（Microcomputer）、微型计算机系统（Microcomputer System）三个层次，如图 1-6 所示。微型计算机包含了许多种系列、档次、型号的计算机，如 IBM PC、联想开天等。这些计算机的共同特点是体积小，适合放在办公桌上使用，而且每个时刻只能一人使用，因此又称为个人计算机。

1.3.1 微型计算机的主要性能指标

（1）字长

字长是衡量计算机处理能力的重要指标，它是指计算机的 CPU 一次能直接处理的二进制数的位数，字长越大，计算机的处理能力越强。例如，8086、286 的字长是 16 位，386、486、586（奔腾）都是 32 位。而现在已有了 64 位甚至多核的微处理器。

（2）速度

衡量计算机速度常用的有两种指标：

- 运算速度：用每秒钟能执行多少条指令来表示。国际上通用的指标是每秒百万条指令 MIPS（Million Instructions Per Second），通常用于较大的计算机系统。
- 主频：指 CPU 主时钟的工作频率（一秒钟内发生的同步脉冲数），主频越高，说明 CPU 的速度越快，微型计算机都用主频这个指标。主频的单位为兆赫兹（MHz），早期的微机 CPU 主频才几兆赫兹，而现在已达到 3GHz 以上。例如，“酷睿 2/2.4G”，“酷睿 2”表示 CPU 的型号（酷睿双核 CPU），“2.4G”就表示主频是 2.4GHz。

（3）内存容量

通常用字节（Byte）作基本单位。目前，微型计算机的内存容量已达到数 G 字节。

（4）存取周期

微机内主存完成一次读/写操作所需的时间称为存储器的存取时间，连续两次读/写所需的最短时间称为存储器的存取周期。存取周期越短，则存取速度越快。存取周期的大小影响着计算机的运算速度。

（5）外部设备的配置和软件的配备

要求显示器、键盘、打印机、磁盘和光盘驱动器等外部设备基本配置齐全，安装合理、达到测试标准。对操作系统、汉字处理能力、数据库管理系统以及网络功能等必须作全面衡量。

此外，还有性能价格比、可靠性、可用性、可维护性等指标。

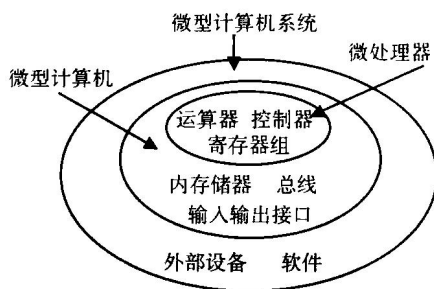


图 1-6 微型计算机系统的组成

1.3.2 微型计算机的硬件组成

微型计算机的硬件由主机和外部设备组成，图 1-7 是 IBM PC 系列机的典型结构。

1. 主板

主板（Main Board）是固定在计算机主机箱箱体上的一块电路板（如图 1-8 所示）。主板上装有大量的有源电子元件。其中主要组件有 CMOS、基本输入输出系统（BIOS）、高速缓冲存储器（Cache）、内存插槽、CPU 插槽、键盘接口、硬盘驱动器接口、总线扩展插槽（提供

ISA、PCI 等扩展插槽)、串行接口 (COM1、COM2)、并行接口 (打印机接口 LPT1)、通用串行总线 USB 接口等, 因此, 主板是计算机各种部件相互连接的纽带和桥梁。

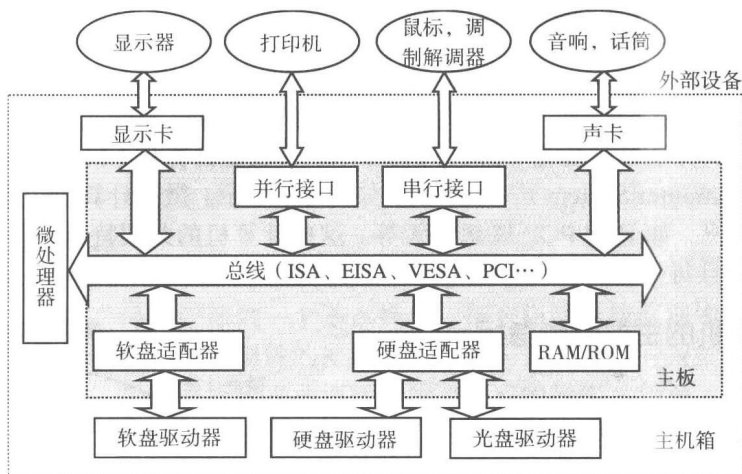


图 1-7 PC 的典型结构

2. 中央处理器

中央处理器 (Central Processing Unit) 简称 CPU (见图 1-9), 是计算机的核心, 包括运算器、控制器等。计算机的运转是在 CPU 的控制下实现的, 所有的算术运算和逻辑运算都是由它完成的。因此, CPU 是决定计算机速度、处理能力和产品档次的关键部件。

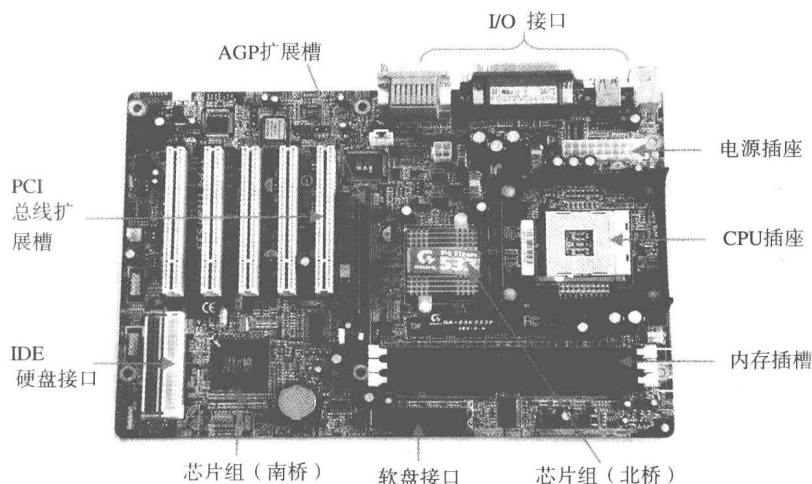


图 1-8 主板



图 1-9 CPU

3. 存储器

存储器 (storage/memory) 分为主存储器 (内部存储器) 和辅助存储器 (外部存储器), 通常称为内存和外存。计算机的存储器层次结构如图 1-10 所示。CPU 只能与内存之间直接进行信息交换, 用户编写的程序以文件的形式存储在外存上, 程序运行时, 装载入内存, 再一条一条装入 CPU 的控制器执行。数据则送入运算器, 运算的结果再送入内存, 然后在屏幕上显示、打印机上打印, 或以数据文件的形式输出到外存保存起来。

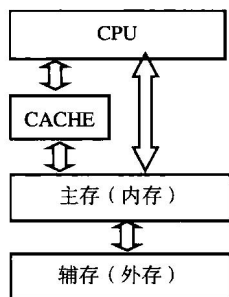


图 1-10 存储器层次结构