

软件工程系列规划教材

C语言程序设计

王仲民 主编

李震雄 易超 康雁 编著

programming
in C



科学出版社

软件工程系列规划教材

C 语言程序设计

王仲民 主编

李震雄 易超 康雁 编著

科学出版社

北 京

内 容 简 介

本书以强调算法应用、问题驱动的程序设计为主线,以软件工程原则为指导思想,以构思-Conceive、设计-Design、实现-Implement、运作-Operate (CDIO)工程能力培养为目标,按照由浅入深、循序渐进的步骤,采取通过程序设计学习C语言的方法安排教学内容。在简要介绍计算机与计算机程序、计算机程序设计与程序语言、CDIO与程序设计的基础上,从软件工程师与计算机的简单对话开始学习程序设计,循序渐进地介绍了面向用户的输入与输出。之后,针对简单问题、决策性问题、重复性问题,以及复杂数据处理(数组、字符串、结构体与联合体、指针等)过程建立解决方案并完成代码实现;以解决方案的拓展与重用的方式介绍了函数设计与实现,以及从程序设计到软件开发的基本方法。核心章节每章有小结和CDIO案例。本书最后简要总结了C语言基本元素。附录中列出了C语言程序常见错误、C语言程序的风格、常用开发环境、标准C库函数和ASCII码表。

本书读者对象主要是大学本科一年级的新生,也可以作为面向应用的设计人员的参考书。

图书在版编目(CIP)数据

C语言程序设计/王仲民主编. —北京:科学出版社,2014.3

软件工程系列规划教材

ISBN 978-7-03-039817-8

I. ①C… II. ①王… III. ①C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(2014)第031362号

责任编辑:于海云 / 责任校对:宣 慧

责任印制:闫 磊 / 封面设计:迷底书装

科 学 出 版 社 出 版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

北京市文林印务有限公司 印刷

科学出版社发行 各地新华书店经销

*

2014年3月第 一 版 开本:787×1092 1/16

2014年3月第一次印刷 印张:17 1/2

字数:459 000

定价:39.00 元

(如有印装质量问题,我社负责调换)

前 言

软件(Software)、程序(Program)、程序语言(Programming Language)、程序设计(Programming)是既相互关联又相互独立的概念。软件包含了若干具有特定功能的程序、与程序相关的数据以及文档等内容。程序是计算机能够系统化完成特定任务的一系列指令。程序设计则是建立解决特定问题具体方案的过程,其核心是将方法、步骤、约束条件、应对策略等应用于原始数据的处理过程,提取并呈现出有价值的完备信息。可见,无论是软件设计或者程序设计都应该与具体的程序实现的语言,也就是一般意义上的程序设计语言相对独立。因为程序语言其实是实现算法或者程序设计思想的具体代码,而代码可以是直接面向计算机硬件的低级语言,如机器语言或汇编语言;也可以是更接近人类自然语言的高级语言,如 C 语言。简言之,正如人类的思想不会受到语言的约束一样,程序设计也应该独立于实现设计思想的计算机语言。

虽然人类的思想独立用于表达思想的具体语言,但是思想的交流却必须以具体的语言为媒介,没有思想的交流就没有人类社会的进步。人类社会所面临的诸多挑战和错综复杂的问题已经不可能仅仅依靠人类自身的处理能力来应对和解决。即使有的问题可能相对简单,但是冗长而繁复的处理过程有可能让解决问题的有利条件随时间而消逝。借助计算机强大的数据处理能力,人类可以实现解决复杂问题的设计思想。但是作为拓展人类能力的工具,计算机如果没有具体的工作指令也将毫无用处。程序语言就是实现人机对话,解决复杂问题从而完成特定任务的关键媒介。

用于实现经缜密设计而形成的问题解决方案的程序语言有很多种,但 C 语言是最为成熟、并且应用广泛的一种高级程序语言。C 语言既能够实现简单设计的程序,也能够开发复杂的软件系统,兼具低级语言的灵活性与高级语言的易用性。其结构化的、面向过程的特征能够体现工程项目自顶向下、逐步分解的系统思维;其模块化的实现方法又能够反映由下向上、构件化扩展解决方案的软件开发原则。掌握 C 语言的基本应用方法可以在一定程度上建立面向对象的软件开发基础,而语言特征的进一步扩展形成了面向对象的程序语言如 C++ 和 Java 等。虽然现代大型软件系统的设计与开发普遍采用的是面向对象的方法,但对于面向工程应用领域的人员而言, C 语言仍然是学习程序设计与代码实现的良好工具。

任何工程项目的实施过程都要经历从构思(Conceive)到设计(Design),从实现(Implement)设计思想到产品正常运行(Operate)的基本过程。CDIO 模式工程教育的目的就是为了使未来的工程师具备良好的分析问题和解决问题的能力,而工程教育领域中任何基础课程或者专业课程都应该是为实现上述目标而设置的。从软件开发或者程序设计的角度而言,任何现代软件产品的开发都必须经过需求获取、模型分析、方案设计、代码实现、质量监控、产品维护等基本过程。其本质是对问题的深入理解和分析,运用领域知识设计解决问题的方案,采用具体的程序语言实现设计思想,在保证软件产品质量的基础上提交、维护软件产品,并在适当时机对解决方案加以扩展以便适应不断变化的应用需求。

本书的编写过程尝试遵循软件工程的基本原则、体现 CDIO 工程教育的基本目标,培养学生分析问题、解决问题的能力,在介绍程序设计基本方法的基础上学习实现设计思想

思想的 C 程序语言。通过问题驱动的方式帮助学生建立基本的程序设计基础，逐步培养分析问题和解决问题的能力；在此基础上适当介绍 C 语言的特殊应用方法，尽可能地避免将程序设计与实现的学习过程演变为研究程序语言的特性、忽略应用目的而专注于程序语言使用技巧的场景中。

本书编写组的作者长期从事程序设计课程的教学和实验指导工作。因此，本书的编写过程可以看作是对作者们在过去若干年为软件学院一年级新生讲授程序设计课程的经验总结。其中，康雁负责第 5 章；易超负责第 1 章、第 2 章、第 3 章；李震雄负责第 6 章、第 9 章、第 10 章、第 11 章；王仲民负责其余章节和附录等内容以及全书的统稿工作。此外，本书内容的组织与编排在一定程度上反映了我们在教学过程中的尝试与探索。我们认为，对于一年级的新生而言，学习正确的程序设计方法比简单地掌握一种程序语言更重要；程序设计的教学过程应该逐步培养学生良好的思维习惯、分析问题和解决问题的能力。为学生建立良好的基础才是程序设计课程和工程教育的根本目的。

我们的同事何婧参与了本书编写初期的讨论，特此致谢。本书的编写还得到了云南大学教务处、软件学院领导、软件学院实验中心及相关教师的支持，在此一并致谢。

由于时间、水平有限，书中难免存在疏漏和不妥之处，敬请读者批评指正。

编 者

2013 年 11 月

目 录

前言

第一篇 程序设计基础

第 1 章 计算机与计算机程序	3
1.1 硬件与软件	3
1.1.1 计算机的基本组成	3
1.1.2 计算机的基本原理	5
1.2 计算机数据处理技术	6
1.2.1 字符及文本处理	7
1.2.2 数字媒体数据处理	8
1.2.3 现代计算	9
1.2.4 云计算的关键	9
思考与练习	9
第 2 章 计算机程序设计与程序语言	10
2.1 程序设计与软件开发	10
2.1.1 程序设计语言的分类	10
2.1.2 程序设计	11
2.1.3 软件开发	11
2.2 程序设计与算法	13
2.2.1 算法及其特征	13
2.2.2 算法的表示	14
2.3 程序设计的实现与程序语言	15
2.3.1 编程语言	15
2.3.2 编码的风格	16
2.3.3 编程准则	17
思考与练习	17
第 3 章 CDIO 与程序设计	18
3.1 CDIO 基本内容	18
3.2 软件工程基本原则	18
3.2.1 软件工程的基本原理	18
3.2.2 以 CDIO 方法体现软件工程的原则	20
3.2.3 面向过程与面向对象	20
思考与练习	21

第二篇 问题驱动的程序设计

第4章 软件工程师与计算机的简单对话	25
4.1 数据处理	25
4.1.1 数据处理简化流程	25
4.1.2 简单问题求解	27
4.2 数据类型与变量	29
4.2.1 数据类型	29
4.2.2 变量定义	30
4.2.3 算术运算与赋值运算	31
4.2.4 数据类型的转换	33
4.3 小结	33
4.4 CDIO 应用案例	34
思考与练习	35
第5章 面向用户的输入与输出	36
5.1 输入处理	36
5.1.1 单字符输入	36
5.1.2 字符串输入	37
5.1.3 格式化输入	37
5.1.4 文件读取	39
5.1.5 文件的打开与关闭	39
5.2 输出处理	43
5.2.1 单字符输出	43
5.2.2 字符串输出	44
5.2.3 格式化输出函数	44
5.2.4 格式字符串	45
5.2.5 文件写	47
5.2.6 格式化写文件	49
5.3 用户界面设计初步	50
5.4 小结	51
5.5 CDIO 应用案例	51
思考与练习	56
第6章 简单问题	58
6.1 算法的直接应用	58
6.1.1 简单问题的需求规约	58
6.1.2 简单问题的算法选择	59
6.2 位运算	61
6.2.1 位运算	61
6.2.2 & 按位与运算	62

6.2.3	按位或运算	63
6.2.4	^ 按位异或运算	63
6.2.5	~按位取反运算	64
6.2.6	不同存储空间大小数据的位运算	64
6.2.7	<< 左移运算	65
6.2.8	>> 右移运算	65
6.3	解决方案的拓展问题	66
6.4	小结	70
6.5	CDIO 应用案例	70
	思考与练习	72
第 7 章	决策性问题	73
7.1	关系运算与逻辑运算	73
7.1.1	关系运算	73
7.1.2	逻辑运算	75
7.2	边界型条件 if 以及 if-else	76
7.2.1	if 与 if-else 分支结构	76
7.2.2	多因素约束条件与嵌套式 if-else 结构	80
7.2.3	条件运算符 ?:	86
7.3	预设型条件 switch-case-break	87
7.4	综合性条件的决策	89
7.5	小结	90
7.6	CDIO 应用案例	92
	思考与练习	95
第 8 章	重复性问题	96
8.1	有限次重复与 for 循环	96
8.1.1	计数循环过程设计及 for 语句的使用	96
8.1.2	嵌套式 for 语句的使用方法	99
8.2	条件重复与 while 和 do-while 循环	101
8.2.1	先验条件循环设计与 while 语句	101
8.2.2	后验条件循环设计与 do-while 语句	103
8.3	for 循环与 while 循环的选择与综合应用	105
8.3.1	循环次数省略与 continue 语句	106
8.3.2	循环状态终止与 break 语句	107
8.3.3	特殊循环和跳转与 goto 语句	109
8.3.4	for 循环的特殊应用	111
8.3.5	循环结构的特殊用例	113
8.4	小结	115
8.5	CDIO 应用案例	116
	思考与练习	120

第 9 章 复杂数据中相同数据类型的处理	121
9.1 一维数组	121
9.1.1 使用数组的原因	121
9.1.2 一维数组的声明和定义方法	122
9.1.3 一维数组的初始化方法	122
9.1.4 一维数组的元素访问方法	123
9.1.5 一维数组的普通应用	124
9.1.6 一维数组在模块化编程中的应用	126
9.2 二维数组	129
9.2.1 使用二维数组的原因	129
9.2.2 二维数组的声明和定义方法	129
9.2.3 二维数组的初始化方法	130
9.2.4 数组完整初始化的略写方法	131
9.2.5 二维数组元素访问方法	132
9.2.6 二维数组的普通应用	133
9.2.7 二维数组在模块化编程中的应用	134
9.3 三维及多维数组	135
9.3.1 三维数组的声明和定义方法	136
9.3.2 三维数组的初始化方法	136
9.3.3 三维数组的元素访问方法	138
9.3.4 三维数组的普通应用	139
9.3.5 三维数组在模块化编程中的应用	141
9.4 字符串	142
9.4.1 字符串的输入与输出	142
9.4.2 字符数组的声明和初始化	143
9.4.3 字符串的访问方法	143
9.4.4 字符串的普通应用	144
9.4.5 字符串在模块化编程中的应用	149
9.5 小结	150
9.6 CDIO 应用案例	150
9.6.1 案例 1: 整型数据的输入校验	150
9.6.2 案例 2: 单词统计	154
思考与练习	157
第 10 章 复杂数据中不同数据类型的处理	159
10.1 结构体	159
10.1.1 使用结构体的原因	159
10.1.2 结构体的定义和声明	159
10.1.3 结构体的成员访问和初始化	161
10.1.4 结构体的简单应用	162

10.1.5	结构体在模块化编程中的应用	162
10.2	结构体数组	163
10.2.1	使用结构体数组的原因	163
10.2.2	结构体数组的声明和初始化	164
10.2.3	结构体数组的应用	165
10.2.4	结构体数组关联数组	166
10.3	联合体	166
10.3.1	使用联合体的原因	166
10.3.2	联合体的定义和声明	167
10.3.3	联合体的成员访问和初始化	168
10.3.4	联合体的简单应用	169
10.3.5	联合体在模块化编程中的应用	170
10.4	枚举数据类型	170
10.4.1	使用枚举数据类型的原因	170
10.4.2	枚举数据类型的定义和声明	170
10.4.3	枚举数据类型的常量列表和常量值	171
10.4.4	枚举数据类型的简单应用	171
10.5	小结	172
10.6	CDIO 应用案例	172
	思考与练习	175
第 11 章	数据的间接访问	176
11.1	指针	176
11.1.1	数据的访问方式	176
11.1.2	指针的声明与使用	177
11.1.3	指针的运算	179
11.2	指针与传地址	180
11.2.1	函数的传值调用	180
11.2.2	函数的传地址调用	181
11.3	数组与指针	182
11.3.1	数组名就是指针常量	182
11.3.2	数组元素的指针访问形式	183
11.3.3	数组名作为参数时	184
11.3.4	数组名形式参数的等价声明	185
11.3.5	字符数组与字符指针	186
11.4	函数与指针	186
11.4.1	指向函数的指针声明和定义	186
11.4.2	指向函数的指针初始化和指针应用	187
11.4.3	指向函数的指针与返回值是指针的函数	187
11.4.4	复杂声明	187

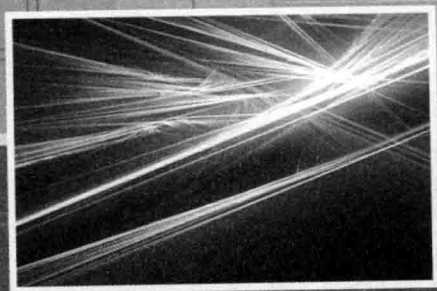
11.5	结构体与指针	188
11.5.1	指向结构体的指针	188
11.5.2	通过指针访问结构体成员	188
11.5.3	自引用结构体	189
11.5.4	简单的链表	189
11.6	小结	190
11.7	CDIO 应用案例	190
11.7.1	案例 1: 整型数据的输入校验	190
11.7.2	案例 2: 单词统计	194
11.7.3	案例 3: 根据输入值给出对应月份值及对应英文名称	197
	思考与练习	198
第 12 章	解决方案的拓展与重用: 函数	199
12.1	模块重用与函数	199
12.1.1	软件重用基本概念	199
12.1.2	程序的模块化设计	200
12.1.3	函数及其定义	201
12.1.4	函数调用方法及参数传递	205
12.1.5	函数及变量的作用域	211
12.1.6	递归函数	216
12.2	标准库函数与自建函数库	219
12.2.1	标准库函数	219
12.2.2	自建函数库	223
12.3	小结	224
12.4	CDIO 应用案例	226
12.4.1	案例 1: 一元二次方程求解	226
12.4.2	案例 2: 汉诺塔问题	229
	思考与练习	231
第 13 章	从程序设计到软件开发	232
13.1	模块化程序与信息隐藏	232
13.1.1	多模块编译	232
13.1.2	自定义头文件	234
13.2	结构化与面向对象程序设计方法	235
13.3	小结	236
	思考与练习	236

第三篇 C 语言参考

第 14 章	C 语言基本元素	239
14.1	基本数据类型及算符	239
14.1.1	基本数据类型	239

14.1.2	算符	240
14.1.3	运算符的优先顺序	240
14.2	流程控制	241
14.2.1	if 和 if-else 语句以及条件表达式	241
14.2.2	switch 语句	243
14.2.3	for 循环	243
14.2.4	while 循环	244
14.2.5	break 与 continue	244
14.3	复杂数据类型	245
14.3.1	数组与字符串	245
14.3.2	地址与指针	245
14.3.3	结构体及结构体数组	245
14.3.4	联合体	246
14.3.5	枚举	246
14.4	函数	246
14.4.1	标准库函数	246
14.4.2	自建函数	247
14.4.3	函数调用方法及返回值	247
14.5	文件	248
14.5.1	文本文件与二进制文件	248
14.5.2	文件的基本操作	248
14.6	预处理	249
14.6.1	宏定义	249
14.6.2	文件包含	249
14.6.3	条件编译	250
附录 A	C 语言程序常见错误	251
附录 B	C 语言程序的风格	252
附录 C	C 语言常用开发环境	253
C.1	用 Microsoft Visual C++ 2008 速成版编写 C 语言程序	253
C.2	用 Dev-C++ 编写 C 语言程序	256
附录 D	标准 C 库函数	260
附录 E	ASCII 码表	264

第一篇 程序设计基础



本篇包括 1~3 章，尝试建立程序设计的基础。

第 1 章 计算机与计算机程序

简要介绍计算机的基本组成、计算机的基本原理及计算机数据处理方式。

第 2 章 计算机程序设计与程序语言

介绍程序设计与软件开发、程序设计与算法、程序实现与程序语言等基本概念。

第 3 章 CDIO 与程序设计

简要介绍 CDIO 基本内容及其主要思想、软件工程基本概念，以及面向过程与面向对象程序设计主要思想。

第 1 章 计算机与计算机程序

计算机对我们来说已经不陌生了, 计算机设备已经进入普通百姓家, 那我们对计算机又有哪些了解呢? 我们知道世界上第一台电子计算机是什么时候问世的吗? 计算机硬件系统和软件系统分别指的是什么? “冯·诺依曼机”是什么? 计算机的工作机制和工作原理是什么? 对于多种类型的各样数据, 计算机是怎么样处理的? 本章将逐一解答这些问题, 带你进入一个计算机的世界。

1.1 硬件与软件

自从世界上第一台电子计算机 ENIAC 于 1946 年问世以来, 电子计算机更新换代, 发展迅猛。20 世纪 70 年代以来, 由于大规模集成电路的发展, 成千上万个晶体管电路可以集成在面积只有几平方厘米的一块电路芯片上, 使计算机的发展进入了飞速时代。目前, 以计算机为代表的信息技术在世界新技术中处于核心和先导地位, 广泛渗透于其他各个新技术领域, 是当今科技发展的重要依托。计算机自诞生以来对人类社会产生了巨大影响。它改变了人类的生产和生活方式, 迅速提高了生产效率, 极大地缩短了时空距离, 有力地推动了经济和社会的全球化。

1.1.1 计算机的基本组成

计算机是一个包括硬件和软件两个部分的复杂系统, 硬件、软件缺一不可。通俗地说, 硬件就是使用的物质对象, 软件解决硬件如何使用的问题。要让计算机能正常工作, 首先必须有机器, 机器系统称为硬件, 这是使用的前提条件。只有硬件还不够, 还必须配置指挥机器进行各种操作时所应遵循的各种程序, 这种程序系统称为软件。

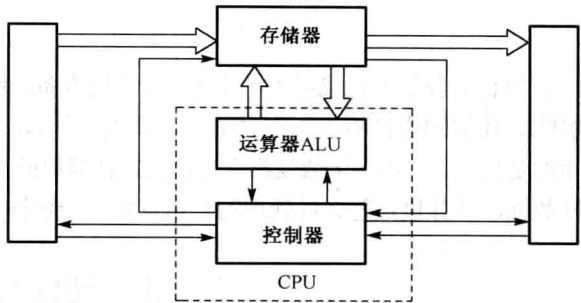
1. 硬件系统概述

硬件系统是构成计算机的物理装置, 指在计算机中看得见、摸得着的有形实体。计算机的硬件由主机和外设组成, 主机由 CPU、内存储器、主板(总线系统)构成, 外部设备由输入设备(如键盘、鼠标等)、外存储器(如光盘、硬盘、U 盘等)、输出设备(如显示器、打印机等)组成。计算机硬件结构如图 1-1 所示。

直到目前, 我们所使用的计算机硬件系统的结构一直沿用由美国数学家冯·诺依曼(John von Neumann)提出的串行工作模型, 它由运算器、控制器、存储器、输入设备、输出设备五大功能部件组成, 并描述了这五个部分的职能和相互关系。这就是“冯·诺依曼机”, 具备了两个本质特点: 一是数据处理采用二进制; 二是提出了“存储程序”的设计思想, 即用记忆数据的同一装置存储执行运算的命令, 使程序的执行可自动地从一条指令进入到下一条指令。这个概念被誉为计算机史上的一个里程碑。存储程序和程序控制原理被称为冯·诺依曼原理。直至今, 计算机虽然已发展成为一个庞大的家族, 其中的每个成员, 尽管在规模、性能、结构和应用等方面存在着很大的差别, 但是它们的基本结构是相同的, 仍然采用存储程序和程序控制的冯·诺依曼原理。

冯·诺依曼原理的基本内容是: 各种各样的信息, 通过输入设备, 进入计算机的存储器,

然后送到运算器，运算完毕把结果送到存储器存储，最后通过输出设备显示出来。整个过程由控制器进行控制。整个工作过程及基本硬件结构如图 1-2 所示。



现代计算机的五个基本组成部分包括：控制器、运算器、存储器、输入设备、输出设备。

1) 控制器

控制器是计算机的“中枢神经”。它主要负责从主存储器中提取指令，对它加以分析，产生相应的控制信号，发送给计算机的各个部分，从而控制计算机各个部分的运行及数据流的正确流动。

2) 运算器

运算器是用来完成算术运算、逻辑运算、测试及其他操作的部件，任何解题中的计算工作都是由它来承担的。它是在控制器的管理下，把需要处理的数据从存储器(或寄存器)中取出，使之流动通过运算器，产生相应的运算结果，随即把运算结果送回存储器(或寄存器)中。数据的每次这样的流动都产生一个运算结果。一般常说的中央处理器(CPU)包括控制器和运算器两大部件。

3) 存储器

主要由高速缓冲存储器(Cache)、主存储器(Main memory)、外存储器所组成的多级(层)存储器系统，是计算机中用于存储程序和数据的记忆装置。这三级存储器各自的功能分工、所用的存储介质的工作原理和特性各不相同。

(1) 高速缓冲存储器(Cache)：在计算机存储系统的层次结构中，介于中央处理器和主存储器之间的高速小容量存储器。它和主存储器一起构成一级的存储器。某些机器甚至有二级三级缓存，每级缓存比前一级缓存速度慢且容量大。这时，一开始的高速小容量存储器就被称为一级缓存。

(2) 主存储器(Main Memory)简称主存：是计算机硬件的一个重要部件，其作用是存放指令和数据，并能由中央处理器(CPU)直接随机存取。主存储器是按地址存放信息的，32 位(bit)的地址最大能表达 4GB 的存储器地址。

(3) 外存储器是指除计算机内存及 CPU 缓存以外的存储器，此类存储器一般断电后仍然能保存数据。常见的外存储器有硬盘、软盘、光盘、U 盘等。使用最多的是磁表面存储器和光存储器两大类，磁表面存储器是将磁性材料沉积在盘片基体上形成记录介质，并在磁头与记录介质的相对运动中存取信息。用于计算机系统的光存储器主要是光盘(Optical Disk)，通常称为 CD(Compact Disk)。目前外存储器的容量不断增大，从 MB 级到 GB 级，甚至到 TB 级海量存储器。

4) 输入设备

输入设备是向计算机中送入程序和数据的有一定独立功能的设备，通过接口和总线与计算机主机连通，用于人-机交互联系，如计算机键盘和鼠标等。

5) 输出设备

输出设备是将计算机处理后的信息以人们能够识别的形式(如文字、图形、数值、声音等)进行显示和输出的设备。常用的输出设备有显示器、打印机、绘图仪等。由于输入/输出设备大多是机电装置,有机械传动或物理移位等动作过程,相对而言,输入/输出设备是计算机系统中运转速度较慢的部件。

2. 软件系统概述

伴随着计算机硬件的不断更新换代,计算机软件也有了很大的发展。软件系统是指使用计算机所运行的全部程序的总称。软件是计算机的灵魂,是发挥计算机功能的关键。它在用户和计算机之间架起了桥梁的作用,给用户的操作带来极大的方便。正是有了内容丰富、种类繁多的软件,使用户面对的不仅是一台实实在在的计算机,而且还包含许多软件的抽象的逻辑计算机(称为虚拟机)。这样,人们可以采用更加灵活、方便、有效的手段使用计算机。软件可分为两大类:一类是系统软件,另一类是应用软件。软件系统组成如图 1-3 所示。

系统软件是管理、监控和维护计算机资源的软件,是用来扩大计算机的功能,提高计算机的工作效率,方便用户使用计算机的软件。系统软件是计算机正常运转所不可缺少的,是硬件与软件的接口。一般情况下系统软件分为 4 类:操作系统、语言处理程序、数据库管理系统和服务程序。

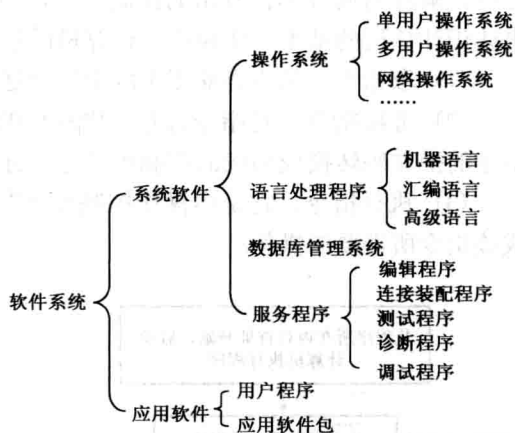


图 1-3 软件系统的组成

应用软件是为了解决计算机各类问题而编写的程序。它分为应用软件包与用户程序。它是在硬件和系统软件的支持下,面向具体问题和具体用户的软件。随着计算机应用的日益广泛深入,各种应用软件的数量不断增加,质量日趋完善,使用更加方便灵活,通用性越来越强。有些软件已逐步标准化、模块化,形成了解决某类典型问题的较通用的软件,这些软件称为应用软件包(Package)。它们通常是由专业软件人员精心设计的,为广大用户提供方便、易学、易用的应用程序,帮助用户完成各种各样的工作。目前常用的软件包有字处理软件、表处理软件、会计电算化软件、绘图软件、运筹学软件等。

系统软件和应用软件之间并不存在明显的界限。随着计算机技术的发展,各种各样的应用软件中有了许多共同的东西,把这些共同的部分抽取出来,形成一个通用软件,它就逐渐成为系统软件了。

1.1.2 计算机的基本原理

1. 计算机的指令系统

指令是能被计算机识别并执行的二进制代码,它规定了计算机能完成的某一种操作。一条指令通常由操作码和操作数两个部分组成:

