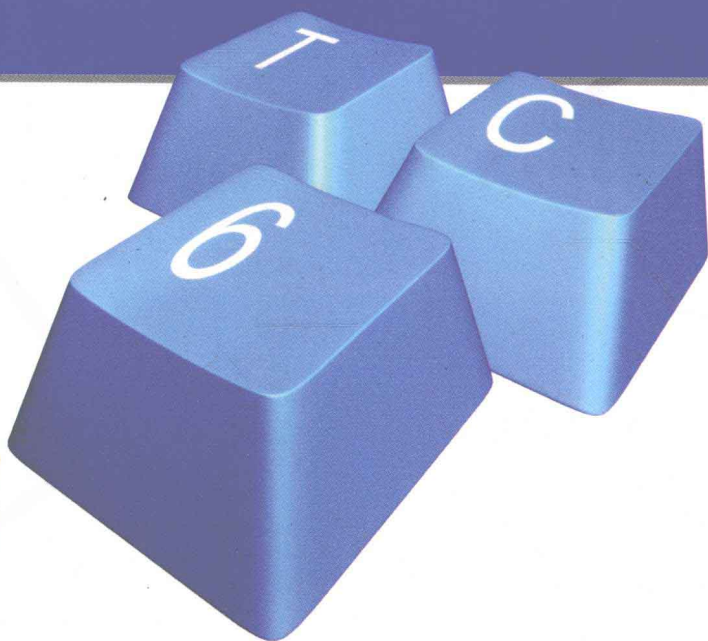


21 世纪面向工程应用型
计算机人才培养规划教材

李德杰 任春美 张丽娜 乔淑云 主 编
李德臣 副主编
王建颖 主 审

C 语言程序设计



清华大学出版社

21 世纪面向工程应用型计算机人才培养规划教材

C 语言程序设计

乔淑云 主 编
李德杰 任春美 张丽娜 李德臣 副主编
王建颖 主 审

清华大学出版社
北 京

内 容 简 介

本书是编者在多次编写讲义、辅导资料、实验指导书的基础上,总结丰富的教学经验并参考大量文献资料编写而成的,主要内容包括程序设计的基本概念、C 语言的特点及数据类型、表达式、运算符、结构化程序设计、函数、数组、指针、结构体、共用体、枚举、用户自定义类型、预处理、文件。各章都配有案例、实验和习题,附录提供了习题参考答案、C 语言常用函数和常见编译错误。还配有课程设计案例及经典的课程设计项目。

本书把培养应用创新能力的目标贯穿于始终,取材新颖,案例丰富,以“实用、以用促学”为编写原则,注重培养程序设计的思路、方法及良好的程序设计风格和习惯,提高解决实际问题的能力。

本书适合作为高等院校各专业“C 语言程序设计”课程的教材、计算机培训班的教材、自学用书,也可作为计算机应用开发人员及普通读者的参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C 语言程序设计/乔淑云主编. —北京:清华大学出版社,2011.12

(21 世纪面向工程应用型计算机人才培养规划教材)

ISBN 978-7-302-27370-7

I. ①C… II. ①乔… 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(2011)第 237865 号

责任编辑:付弘宇 张为民

责任校对:梁 毅

责任印制:李红英

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:北京密云胶印厂

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185×260 印 张:23.5 字 数:584 千字

版 次:2011 年 12 月第 1 版 印 次:2011 年 12 月第 1 次印刷

印 数:1~3000

定 价:36.00 元

产品编号:041561-01

前言

C 语言是当今世界上广泛流行的一种结构化程序设计语言,它既具有高级语言程序设计的特点,又具有汇编语言的功能,能对硬件直接进行操作。它不仅是开发系统软件的常用程序设计语言,而且是开发应用软件和进行大规模科学计算的常用程序设计语言。

“C 语言程序设计”是高等学校普遍开设的一门计算机基础课程,本课程教学存在的普遍现象是:当学习“C 语言程序设计”的知识时感到并不难,但在应用这些知识来解决实际问题时却发现力不从心,不仅不会使用目前流行的 C 开发工具编写程序,而且在考试时往往感到茫然而不知所措,也导致了实际教学效果与教学目标存在相当大的差距。为什么会产生这种“一看就懂,一用就不会”的现象?在培养学生的抽象思维能力、程序设计实践能力上到底要求学生应该掌握什么?更详细地说,什么样的知识是必须具备的,什么样的知识是可以在未来需要时逐步补充的,什么样的知识又是完全可以忽略的?

为了解决这个问题,本书采用主辅两条线,双色排版,主线用黑色,辅线即“阅读知识”部分用蓝色。以程序设计为主线的主体结构注重研究程序设计的思路、方法与技巧,与程序设计有关的 C 语言语法、规则在潜移默化中得以展开,降低了学习难度;以阅读知识为辅线的从属结构意在补充部分基础知识,培养学习兴趣、提高自主学习能力和独立思考的能力,以挖掘读者的潜能。

本书作者长期从事 C 语言程序设计的一线教学、研究和软件开发工作,了解初学者学习 C 语言的难点,在多次编写讲义、辅导资料、实验指导书的基础上积累并拥有丰富的教学经验。本书概念清晰、结构合理、内容严谨、讲解透彻、重点突出、示例典型、实用性强。全书共 11 章和 6 个附录,主要内容包括程序设计的基本概念、C 语言的特点及基本数据类型、表达式、运算符、顺序结构、分支结构、循环结构、函数、数组、指针、链表、结构体、共用体、枚举、用户自定义类型、预处理命令和文件,强调在理论学习的同时加强实践训练。每章都配有实验,每个实验均包括实验目的、实验内容、实验习题,每章还配有课程设计案例及经典的课程设计项目,这些项目难易程度不同,简单的项目可以一个人独立完成,复杂的项目可由几个人共同完成。

教师可采取多种方式使用本教材,在讲授时可以根据学生的背景知识及给定的学时数来进行内容的取舍。

本书由徐州工程学院乔淑云任主编并负责策划、统稿,徐州工程学院李德杰、任春美、张丽娜和中国矿业大学李德臣任副主编,徐州工程学院袁媛、武长柱参与编写,徐州工程学院王建颖任主审。其中,乔淑云编写第 1、9 章,张丽娜编写第 2、3、4 章,任春美编写第 5、6 章,李德杰编写第 7、8 章,乔淑云和袁媛编写第 10 章,乔淑云、李德臣和武长柱编写第 11 章,武长柱编写附录。另外,本书在编写和出版过程中,得到了徐州工程学院信电工程学院邵晓根教授、鲍蓉教授的大力支持,还参阅并引用了一些文献的研究成果,有的收录在本书参考文献

献中,在此一并表示衷心的感谢!

本书是江苏省现代教育技术研究课题,徐州工程学院教学研究课题及重点课程建设项目。由于作者学识水平有限,书中的疏漏和错误之处在所难免,恳请同行专家和读者不吝赐教,在将来修订本书时作为重要的参考,也便于作者提高水平,我们的电子信箱是 qsy@xzit.edu.cn。

作 者

2011 年 12 月

第 1 章 C 语言与程序设计	1
1.1 程序与程序设计	1
1.1.1 程序	2
1.1.2 算法与数据结构	2
1.1.3 程序设计	4
1.1.4 程序设计语言	4
1.1.5 程序设计方法	4
1.2 C 语言概述	5
1.2.1 C 语言简史	5
1.2.2 C 语言的特点	5
1.2.3 C 语言程序结构	6
1.2.4 C 语言编程风格	9
1.2.5 C 程序设计思维	10
1.2.6 C 程序的实现流程	11
1.3 上机环境	12
1.3.1 Turbo C 2.0 开发环境	12
1.3.2 Microsoft Visual C++ 6.0 集成开发环境	14
1.4 案例：高速公路收费站应用程序界面设计	16
1.5 初学者常见问题	17
本章小结	18
实验 1	19
习题 1	19
第 2 章 顺序结构程序设计	21
2.1 数据类型	21
2.1.1 标识符	22
2.1.2 常量与变量	22
2.1.3 数据类型转换	26
2.2 运算符与表达式	27
2.2.1 算术运算符和算术表达式	28
2.2.2 赋值运算符和赋值表达式	30
2.2.3 逗号运算符和逗号表达式	32

2.2.4 运算符优先级和结合性	33
2.3 基本语句	34
2.4 数据输入输出	36
2.4.1 字符数据输入输出	36
2.4.2 格式输入与输出	38
2.5 案例:实现两个变量的数据交换	43
2.6 常见错误	44
本章小结	45
实验 2	45
习题 2	47
第 3 章 选择结构程序设计	50
3.1 关系运算符和表达式	50
3.1.1 关系运算符及其优先级	50
3.1.2 关系表达式	51
3.2 逻辑运算符和表达式	52
3.2.1 逻辑运算符及其优先级	52
3.2.2 逻辑表达式	52
3.2.3 逻辑运算表达式的值	52
3.3 if 语句	54
3.3.1 if 语句的 3 种形式	54
3.3.2 if 语句的嵌套	57
3.3.3 条件运算符和条件表达式	58
3.4 switch 语句	60
3.5 案例:利用 BMI 公式计算是否超重	61
3.6 常见错误	62
本章小结	62
实验 3	63
习题 3	65
第 4 章 循环结构程序设计	67
4.1 循环结构程序概述	67
4.2 goto 语句	67
4.3 while 语句	68
4.4 do-while 语句	69
4.5 for 语句	71
4.6 循环的嵌套	74
4.7 几种循环的比较	77
4.8 break 和 continue 语句	77

4.8.1	break 语句	78
4.8.2	continue 语句	78
4.9	案例: 判定素数	79
4.10	常见错误	81
本章小结		81
实验 4		82
习题 4		83
第 5 章	数组	87
5.1	概述	87
5.2	一维数组	88
5.2.1	一维数组的定义	88
5.2.2	一维数组在内存中的存储	90
5.2.3	一维数组的引用	91
5.2.4	一维数组的初始化	94
5.2.5	一维数组程序举例	96
5.3	二维数组	100
5.3.1	二维数组的定义	100
5.3.2	二维数组的理解	100
5.3.3	二维数组在内存中的存储	101
5.3.4	二维数组的引用	101
5.3.5	二维数组的初始化	102
5.3.6	二维数组程序举例	104
5.4	字符串和字符数组	107
5.4.1	字符数组的定义	107
5.4.2	字符数组的初始化	107
5.4.3	字符数组的引用	108
5.4.4	字符串	109
5.4.5	字符串的输入输出	111
5.4.6	常用字符串处理函数	113
5.4.7	字符数组程序举例	118
5.5	案例: 模拟实现计算器中的进制转换	120
5.6	常见错误	121
本章小结		123
实验 5		125
习题 5		125
第 6 章	函数	130
6.1	概述	130

6.2	函数的定义	133
6.2.1	无参函数定义的一般形式	133
6.2.2	有参函数的一般形式	133
6.3	函数的调用	135
6.3.1	调用的一般形式与方式	135
6.3.2	参数传递和函数的返回值	136
6.3.3	对被调用函数的声明	140
6.4	函数的嵌套与递归	142
6.4.1	函数的嵌套调用	142
6.4.2	函数的递归调用	143
6.5	数组作函数参数	145
6.5.1	数组元素作函数参数	145
6.5.2	数组名作函数参数	146
6.6	局部变量和全局变量	153
6.6.1	局部变量	153
6.6.2	全局变量	155
6.7	变量的存储类别	158
6.8	案例：寻找黑色星期五	162
6.9	常见错误	164
	本章小结	165
	实验 6	166
	习题 6	168
第 7 章	指针	173
7.1	基本概念	173
7.1.1	内存地址	173
7.1.2	变量地址	174
7.1.3	存取变量值	174
7.2	指针变量的定义与引用	175
7.2.1	指针变量的定义	175
7.2.2	指针变量的初始化与引用	176
7.2.3	指针运算	179
7.2.4	空指针和 void 型指针	182
7.2.5	二级指针	183
7.3	指针与数组	184
7.3.1	指针与一维数组	185
7.3.2	通过指针引用数组元素	185
7.3.3	数组名作函数参数	187
7.3.4	指针与二维数组	188

7.4 指针与字符串	192
7.4.1 指针处理字符串	192
7.4.2 字符串的指针作函数参数	194
7.5 指针与函数	195
7.5.1 指向函数的指针变量	195
7.5.2 指针变量作函数参数	196
7.5.3 指针型函数	198
7.6 指针数组和 main 函数的参数	200
7.6.1 指针数组	200
7.6.2 指针数组作函数参数	202
7.7 案例:统计学生成绩	204
7.8 常见错误	207
本章小结	207
实验 7	208
习题 7	209
第 8 章 结构体、共用体与枚举	213
8.1 结构体	213
8.1.1 结构体类型的定义	213
8.1.2 结构体类型变量	215
8.2 结构体数组	221
8.3 结构体与指针	222
8.3.1 指向结构变量的指针	222
8.3.2 指向结构数组的指针	224
8.3.3 结构体指针变量作函数参数	226
8.4 链表	228
8.4.1 动态存储分配	228
8.4.2 链表的概念	231
8.4.3 链表的基本操作	233
8.5 共用体	239
8.5.1 共用体定义	239
8.5.2 共用体与结构体的嵌套使用	240
8.6 枚举类型	241
8.6.1 枚举类型的定义和枚举变量的说明	242
8.6.2 枚举类型变量的赋值和使用	242
8.7 类型定义符 typedef	244
8.8 案例:选票统计	245
8.9 常见错误	246
本章小结	246

实验 8	247
习题 8	248
第 9 章 编译预处理	253
9.1 预处理概述	253
9.2 宏定义	254
9.2.1 无参宏定义	254
9.2.2 有参宏定义	256
9.2.3 宏嵌套	258
9.3 文件包含与条件编译	259
9.3.1 文件包含	259
9.3.2 条件编译	261
9.4 案例：系统可移植性处理	262
9.5 常见错误	263
本章小结	264
实验 9	265
习题 9	265
第 10 章 文件	268
10.1 文件概述	268
10.1.1 文件概念	268
10.1.2 文件类型	269
10.1.3 文件指针	270
10.1.4 文件的打开和关闭	271
10.1.5 文件的操作顺序	273
10.2 文件的读写操作	274
10.2.1 字符读写函数 fgetc()和 fputc()	274
10.2.2 字符串读写函数 fgets()和 fputs()	276
10.2.3 数据块读写函数 fread()和 fwrite()	277
10.2.4 格式化读写函数 fscanf()和 fprintf()	278
10.3 文件的定位	279
10.3.1 文件指针位置定位函数 fseek()	279
10.3.2 文件指针位置复位函数 rewind()	280
10.3.3 文件指针位置查询函数 ftell()	280
10.4 文件检测函数	281
10.4.1 文件结束检测函数 feof()	281
10.4.2 文件出错检测函数 ferror()	281
10.4.3 文件出错标志和文件结束标志置 0 函数 clearerr()	282
10.5 案例：文件操作的综合应用	282

10.6 常见错误	287
本章小结	288
实验 10	289
习题 10	290
第 11 章 C 语言课程设计	294
11.1 课程设计说明	294
11.1.1 课程设计的任务、性质与目的	294
11.1.2 课程设计的基本要求	294
11.1.3 完成课程设计应提交的文档	295
11.2 课程设计案例：通讯录管理系统	295
11.2.1 设计方法一：用函数实现	296
11.2.2 设计方法二：用结构体实现	304
11.2.3 设计方法三：用文件实现	313
11.3 课程设计项目	323
11.3.1 项目一：工资管理系统	323
11.3.2 项目二：贪吃蛇	323
11.3.3 项目三：N 皇后问题	324
11.3.4 项目四：其他题目	325
附录 A C 语言关键字	327
附录 B 运算符的优先级和结合性	328
附录 C C 语言常用函数	330
附录 D C 语言常见编译错误	339
附录 E 实验报告参考格式	351
E1 实验报告要求	351
E2 实验报告样例	352
附录 F 习题参考答案	355
参考文献	361

C语言与程序设计

学习任务与目标

- (1) 了解程序设计的基本概念。
- (2) 掌握程序设计的思想与方法。
- (3) 掌握算法与数据结构在程序设计中的地位与作用。
- (4) 了解 C 语言的历史和特点。
- (5) 掌握 C 程序的基本结构和书写风格。
- (6) 熟悉 C 程序的上机步骤和 Visual C++ 6.0 开发环境。
- (7) 会用 C 语言编写一个简单的程序。

内容摘要

本章介绍程序、程序设计、算法与数据结构的基本概念, C 语言的发展、C 语言的特点、C 语言的组成结构、编程风格, 以及 C 语言程序的执行过程, 给出一个案例: 高速公路收费站应用程序界面设计, 说明用 C 语言编写程序的方法及 C 语言的实用性。简要叙述 C 语言的上机步骤及 Turbo C 2.0 开发环境, 详细叙述 Visual C++ 6.0 集成开发环境, 并且总结了初学者常见的问题, 提醒读者避免犯同类型的错误。

1.1

程序与程序设计

我们所使用的工具深刻影响着我们的思维方式和思维习惯, 进而也深刻地影响着思维能力。计算思维代表着一种普遍的认知和一类普适的技能。美国卡内基·梅隆大学的周以真(Jeannette M. Wing)教授于 2006 年在计算机权威期刊《Communications of the ACM》杂志提出: 计算思维(Computational Thinking)是运用计算机科学的基础概念进行问题求解、系统设计, 以及人类行为理解等一系列思维活动。就问题求解(Problem Solving)而言计算思维与数学思维相似, 建立在计算过程的具体实现和约束之上, 通过程序来控制计算机的全部操作。所以程序设计课程在某种程度上肩负了传播计算思维的责任。

“我的程序设计应该选用何种计算机语言?”这是每一个学习程序设计的初学者所面临的第一个问题。

在众多的程序设计语言中, 业界有句经典的说法: “聪明的程序员选用 Delphi, 真正的程序员使用 C 语言、Visual C++”。这句话可以回答上述问题。Visual C++ 是在 C 语言的基础上发展起来的。目前著名的、应用广泛的三大操作系统 Windows、UNIX 和 Linux 都

是用C语言编写的。C语言是计算机界广泛使用、广受欢迎的程序设计语言。学会程序设计,也就意味着走进了计算机的世界。

1.1.1 程序

为了让计算机能按照人的意图工作,就必须设计各种各样的程序。程序(Program)就是为实现特定目标或解决特定问题,用计算机语言编写的可连续执行并能完成一定任务的指令序列的集合。程序控制计算机该做什么不该做什么,是计算机的主宰者,所有托付计算机去做的事情都能写成程序。假如没有程序,那么计算机什么事情也做不了。

计算机程序通常包括算法和数据两部分内容,可用如下算式描述:

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

1.1.2 算法与数据结构

1. 算法

算法是程序的灵魂,不掌握算法就无法编写出程序,编写程序的过程就是设计算法的过程。

所谓算法,是指完成一个任务所采取的一组明确的、有一定顺序的方法和步骤。计算机进行问题求解的算法大致可分为两类:一类是数值算法,主要用于解决数值求解问题。另一类是非数值算法,主要用于解决需要用分析推理、逻辑推理才能解决的问题,如人工智能中的许多问题以及查找、分类等问题都属于这类算法。

怎样衡量一个算法的正确性呢?一般地,可用算法的如下特征来衡量。

- (1) 有穷性。一个算法必须在有限步骤内实现,且每一步都在有限的时间内完成。
- (2) 确定性。算法的每一个步骤必须有确切的定义,不能存在歧义性。
- (3) 有效性。算法的所有操作都能有效执行。例如,在实数范围内求一个负数的平方根,就是无效的操作。
- (4) 具有初始量。可以有零个或多个输入。有些算法不需要从外界输入数据,例如:求5!。而有些算法则必须输入值,例如:求 $n!$,因这里 n 是未知数,需要从键盘输入具体的值。

(5) 具有一个或多个输出。算法的实现是以得到计算结果为目的,无任何输出的算法无任何意义。

算法常用的描述方法如下:

(1) 自然语言。自然语言就是人们日常交流用的语言,如汉语、英语、数学符号等,通俗易懂,比较符合人们的思维习惯,但在内容描述上容易引起歧义性,不易直接转化为程序,一般用于比较简单的情況描述。

(2) 流程图。流程图也称程序框图或传统流程图,是一个描述程序的控制流程和指令执行情况的有向图,是程序的一种比较直观的表现形式。常用的流程图符号如图1-1所示。

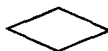
开始、结束框



一般处理框



判断框



输入输出框



流程线



连接符



图1-1 常用的流程图符号

用传统流程图描述算法的优点是流程图可以直接转化为程序,形象直观,各种操作一目了然,不会产生歧义,易于理解和发现算法设计中存在的错误;但缺点是允许用流程线任意转向,可能造成修改和阅读上的困难。

(3) N-S 结构化流程图。N-S 结构化流程图是由美国学者 I. Nassi 和 B. Schneiderman 于 1973 年提出的,并以两位学者名字的首字母命名。它的特点是全部取消了流程线,这样迫使算法只能从上到下顺序执行,从而避免了算法流程的任意转向,保证了程序的质量。用传统流程图和 N-S 结构化流程图描述顺序结构分别如图 1-2 和图 1-3 所示。

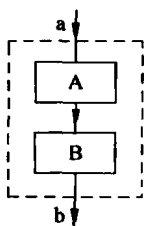


图 1-2 传统流程图

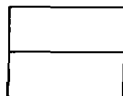


图 1-3 N-S 结构化流程图

(4) 伪代码。伪代码是介于自然语言与计算机语言之间的一种文字和符号的描述方法,其最大优点是计算机语言比较接近,易于转换为计算机程序,书写无固定格式和规范,比较灵活。

【阅读知识 1.1: 算法的歧义性与模糊性】

相传很久很久以前有位老先生对某位后生不满意,做出上联让他对下联:

眉先生,须后生,后生却比先生长。显然这里的“先生”、“后生”都是典型的双关语,具有歧义性。那后生急中生智对出下联:

眼珠子,鼻孔子,珠子反在孔子上。很显然,这里的“珠子”(即朱子,指朱熹)与“孔子”也是双关语,对的绝妙。

在现实生活中,双关语不仅含蓄、幽默,而且有实际意义。然而,在利用计算机编写程序时,计算机没有人类那么善解人意,它只能机械地按照事先定义好的步骤一步一步地执行下去,“先生”到底指人,还是指“先出现”的?如果计算机犯糊涂,麻烦可就大了(会将错误不费力地传播出去)。

此外,算法不允许有模糊性。例如,“拿 2 斤苹果 3 斤梨来”有明确的数量,在程序中是可以操作的,但如果说“拿一点苹果和梨来”,计算机就无所适从了,日常生活中的“一点”在计算机中到底是多少呢?对计算机来说具有模糊性,读者在编写程序时应避免。

2. 数据结构

数据结构是数据的组织方式,包括数据元素、元素间的联系及数据运算。数据结构与算法密不可分,一个好的数据结构能使算法简单化,一个好的算法能明确问题,并能更好地设计数据结构,二者相辅相成。

一般将数据结构分为两大类型:线性结构与非线性结构。栈、队列和串等是线性结构,广义表、树和图等是非线性结构。

1.1.3 程序设计

程序设计(Programming Design)是根据计算机要完成的任务,设计算法,选择适当的数据结构,编写对应的代码,上机测试运行,并能得到正确的结果。通常说的“编程”是“编写程序”的缩写,意思就是“程序设计”。程序设计应遵循规定的原则和方法,可用如下算式描述:

程序设计 = 算法 + 数据结构 + 程序设计方法

1.1.4 程序设计语言

计算机本身是无生命、无逻辑思维的一种工具,不能识别人们日常生活中使用的自然语言。要想使计算机充满“活力”、“为人民服务”,就必须让它执行相应的程序,这些程序都是依靠程序设计语言编写出来的。

程序设计语言(Programming Language)是一系列对计算机进行操作的规则,是人与计算机进行交流的语言工具。像日常生活中使用的语言一样,程序设计语言也具有规定的语法、语义和使用环境。

计算机程序设计语言种类繁多,从其发展及功能角度大致划分为机器语言、汇编语言、面向任务的程序设计语言和面向对象的程序设计语言。

1.1.5 程序设计方法

常用的程序设计方法有 3 种:结构化程序设计、面向对象程序设计和并行程序设计。

结构化程序设计(Structured Programming, SP)的基本思想是采用自顶向下、逐步求精、模块化和单入口/出口的控制结构。结构化程序设计方法遵循 3 个原则:

- (1) 自顶向下逐步求精。
- (2) 模块化设计。
- (3) 使用 3 种基本控制结构:顺序结构、选择结构和循环结构。

面向对象程序设计(Object Oriented Programming, OOP)的基本原则是尽可能模拟现实世界中人类的思维方式,使开发软件的方法和过程尽可能地接近人类解决问题的方法和过程。面向对象程序设计方法具有 6 个特征:对象、类、抽象、继承、封装、消息传递。

并行程序设计(Parallel Programming)比串行程序设计复杂,主要涉及操作系统及优化编译方面的知识。并行程序设计的过程主要划分为 4 个步骤:任务划分、通信分析、任务组合、处理器映射。

【阅读知识 1.2: 程序设计思想前沿】

目前,程序设计的最新发展主要是组件技术和面向切面的编程技术。面向对象技术催生了组件技术。组件技术引入了用于生成业务结果的一套改进的机制,为软件开发提供了改良的方法。组件技术主要包括组件的下层构造、软件模式、软件构架、基于组件的开发等。组件被认为是面向对象和其他软件技术的化身。

面向切面编程(Asspect Oriented Programming, AOP)是一种新的编程技术。作为消除代码重复的一种方法,AOP 弥补了面向对象的编程(OOP)在跨越模块行为上的不足,是 OOP 的延续。AOP 引进了 Aspect 概念,将影响多个类的行为封装到一个可重用模块中,

以利于消除 OOP 引起的代码混乱和分散问题,并增强系统的可维护性和代码的重用性。AOP 包括组件语言(利用传统的面向对象语言,如 Java、C++ 编写系统的核心关注点)、切面语言(AOP 中专门处理切面的语言,用于编写系统的横切关注点),以及编织器(负责对两种语言编写的功能代码进行编织,并形成最终可运行的程序)。

1.2 C 语言概述

C 语言是目前国际上广泛流行的一种结构化程序设计语言,它既具有高级语言程序设计的特点,易学易用,又具有汇编语言的功能,能对硬件直接进行操作。它不仅是开发系统软件的常用程序设计语言,而且也是开发应用软件和进行大规模科学计算的常用程序设计语言。世界上高级语言有上百种,每种语言也都在不断地发展变化。C 语言是大学生学习计算机语言的一种基本选择,也是进一步学习面向对象语言的基础或台阶。

1.2.1 C 语言简史

C 语言是由早期的编程语言 BCPL(Basic Combined Programming Language)发展演变而来的。20 世纪 70 年代初,AT&T 贝尔实验室的 Ken Thompson 根据 BCPL 语言设计出较先进的并取名为 B 的语言,1972 年 Dennis M. Ritchie 在 B 语言的基础上研制了 C 语言,并用 C 语言编写了第一个在 PDP-11 计算机上实现的 UNIX 操作系统。1977 年,K&R 合著《The C Programming Language》一书,标志了 C 语言的诞生。

C 语言的标准:美国国家标准化组织(ANSI)于 1983 年专门成立了 C 语言标准委员会,完成了 C 语言的标准 C89,随后相继出现 C90、C99。

目前可在微机上运行的 C 编译器和开发环境有 Turbo C/C++、GCC、Microsoft Visual C++、Intel C/C++、Borland C++ Builder 等。

1.2.2 C 语言的特点

C 语言之所以流行,主要因其有如下几个特点:

(1) C 语言简练、紧凑,使用方便、灵活,运算符和数据结构丰富。C 语言有 32 个关键字,9 种控制语句,标识符的定义非常灵活,支持大小写敏感。有 34 种运算符,包括算术、赋值和强制类型转换等,数据类型有整型、实型、字符型、数组类型、指针类型、结构类型和共用类型。尤其是指针类型数据,使用十分灵活和多样化。

(2) C 语言兼有高级语言和低级语言的特点。C 语言程序要通过编译、连接才能得到可执行的目标程序,这和其他高级语言是相同的。C 语言允许直接访问物理地址,能进行位(bit)操作,并能实现汇编语言的大部分功能,可以直接对硬件进行操作,既是成功的系统描述语言,又是通用的程序设计语言。

(3) C 语言是一种结构化程序设计语言。支持结构化程序设计的 3 种基本结构:顺序结构、选择结构和循环结构。C 语言程序书写形式自由,一行可以写一条或多条语句,大小写敏感,习惯用小写。

(4) C 语言是一种模块化程序设计语言。C 语言程序由一系列函数构成,函数为独立