



C/C++ 与数据结构

(第2版)

王立柱
编著

清华大学出版社

C/C++ (第2版) 与数据结构

王立柱 编著

清华大学出版社
北京

内 容 简 介

本书不仅是 C, C++ 和数据结构的综合性教材, 也可以单独用作 C 和 C++ 语言教材, 或 C 和 C++ 描述的数据结构教材。本书配有多媒体软件, 既可以助教, 又可以助学。而且内容丰富, 读者可以根据自己的专业特点, 选择相应的内容。

书 名: C/C++ 与数据结构(第 2 版)

作 者: 王立柱 编著

出 版 者: 清华大学出版社

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

社 总 机: 010-62770175

地 址: 北京清华大学学研大厦

邮 编: 100084

客户服务: 010-62776969

责任编辑: 马瑛珺

封面设计: 郑 晶

印 刷 者: 清华大学印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 185×260 印张: 26.25 字数: 601 千字

版 次: 2003 年 8 月第 2 版 2003 年 8 月第 1 次印刷

书 号: ISBN 7-89494-094-1

印 数: 0~6000

定 价: 39.00 元(含光盘)

为了适应计算机科学的飞速发展和社会对计算机综合人才的广泛需求,计算机基础教育必须改革,改革的目标就是综合。这本教材便是这种改革实践 4 年的总结,它将 C 语言、数据结构和 C++ 综合为一门程序设计基础课程,可以不分专业,在大学一年级完成(授课 80 学时,实验 80 学时)。

计算机科学越来越趋向综合,C++ 这门代表着最先进的程序设计方法的语言本身就代表着综合,它的第一个代表性的概念“类”便是对数据和操作的综合。

而综合绝不意味着拼凑、肤浅和庞杂,改革实践表明,综合带来了深入、统一、明确和简约,从而使面向对象的程序设计方法和 C++ 可以面向更广泛的对象,使“数据结构”这门计算机专业最为核心的课程可以成为每一个人学习计算机的基础。我们愿意把改革的成果与读者共享。

第 1 章 机器语言程序简介。在模型计算机指令系统的基础上,以机器语言程序的形式,讲解对程序设计者来说最重要的硬件知识。我们的目的不是要编写机器语言程序,而是因为这些知识随着程序设计的深入,不断在引申、推广、抽象,所以,具体而明确地掌握这些知识,是我们继续学习的基础。

第 2 章 C 语言基础。我们直接从变量和指针、数组和指针、函数和指针的关系入手,讲解 C 语言。指针即地址,它和函数构成 C 语言的核心。传统的方法都是通过变量名引进 C 程序,而尽可能推迟讲解指针,甚至淡化指针,以便减轻理解 C 语言的难度。可是指针是 C 语言的核心,它是理解 C 程序、解决 C 程序问题的关键。

第 3 章 数据结构概论。本章阐述了数据结构、数据类型、算法和程序的概念以及它们之间的关系,并且通过实例介绍程序设计的一般方法。

第 4,5,6 章 分别讲解了顺序表、顺序队列和顺序栈。通过构造新的数据类型,表明我们进入了真正的程序设计阶段,开始了能动地、创造性地学习程序设计。

第 7 章 字符串。本章首先介绍了 C 语言已具备的字符串类型(简称 C 串),然后结合字符串处理中的模式匹配问题,在指出 C 串的局限性的同时,创建新的串类型(简称新串)。

第 8,9 章 分别讲解了链表和链队列。通过与顺序存储结构对比,指出后者的不足,从而构造出新的存储模式——链式存储结构。并综合所学内容,设计了银行窗口服务的模拟程序——事件驱动模拟程序。

第 10 章 二叉树。从本章开始进入非线性部分。二叉树是研究其他非线性结构的

基础。我们先后使用递归和非递归两种方法讲述遍历,并把很多典型的算法作为遍历的直接应用。

第 11 章 树。本章通过具体的遍历算法阐明树和二叉树之间的等价关系;作为树的遍历算法的应用,介绍了八皇后问题的递归和非递归算法;最后,结合八皇后结果的图形输出设计,介绍了用 C 语言作图的基本知识。

第 12 章 图。图是应用最广泛的非线性结构。本章研究并实现了图的典型算法:最小生成树、单源最短路径、拓扑序列和关键路径。

第 13 章 C++ 程序。本章主要用 C++ 描述已有的 C 程序,比如中缀表达式求值、串类等,在对比中学习 C++。

本书的特点是综合:变量和指针,数组与指针,函数与指针是一个综合;C 串与 C++ 串是综合;快速排序和幂集问题归入前序遍历,hanio 问题归于中序遍历,堆排序和哈夫曼树是堆类的直接应用,八皇后问题在树的前序遍历中解决,这是综合;迷宫归于图的层次遍历,骑士巡游隶属图的前序遍历,这还是综合。

本书的创新是从面向对象的高度设计 C 程序,为 C++ 提供充分的感性基础。“C++ 不过是更好的 C”,这在本书中得到了充分的体现。C 语言的不足由 C++ 语言补充和改进,C++ 语言的深奥之处由 C 语言详述。在很多地方,C++ 程序是 C 程序的提炼,C 程序是 C++ 程序的展示。由此,学习 C 语言和学习 C++ 语言成为一个过程。

本书的原则是学习数据结构和学习程序语言共进,它们的中介是算法。语言只有满足算法的需要,才能被认识和掌握。数据结构只有依赖语言的发展才能拓展其应用领域。

总之,每一部分没有独立存在的理由,它们的意义只有在它们的相互依赖、相互补充的关系中得到明确的解答。

正是这种统一性,才产生了既可以助教又可以助学的多媒体软件——配书光盘。正是这种统一性,才使抽象类型和存储类型、算法每一步骤和程序每一行代码、C 程序和 C++ 程序可以同时展现,相得益彰。

我们迈出了改革的第一步——艰难的一步,为此感谢天津师范大学各级领导和同仁一直给予的关怀和支持。

2001 年 8 月

与第 1 版相比,第 2 版教材主要改进部分有:

1. 第 2 章增加了第 14 节“程序设计综合举例”,包含 C 程序设计的典型算法,其中的“起泡排序”、“数制转换”、“Palindrom”、“删除重复数据”、“约瑟夫问题”、“划分”等算法在数据结构部分还有新的描述,学习时可以相互对比。这进一步体现了第一版前言中所阐述的“学习数据结构与学习程序语言共进”的思想。

2. 重新编写了第 2 章第 13 节“文件”,并改名为“流与文件”,增加了第 14 章“C++ 的 I/O 流库”。这两部分内容是一一对应的,分别用 C 语言和 C++ 语言详细描述了对文件的各种运算和处理。为我们在第一版前言中所阐述的“学习 C 语言和学习 C++ 语言成为一个过程”提供了更充实的内容。

3. 在第 1 章第 3 节“子程序调用过程”中加进具体的程序,为学习 C 语言的函数调用,尤其是递归调用,提供了更为具体、形象的感性认识。

4. 第 2 章“基本类型变量和指针”、“一维数组和指针”、“二维数组和指针”、“函数”和“指针参量”等节都做了整体的修改,进一步详述了变量、数组、指针、函数之间的关系:从直接引用的局限性引进了数组;根据模块化程序设计思想,由复合语句引入函数;从函数的形参与实参之间的关系,研究了指针变量与数组之间的匹配问题。

5. 第 2 章第 12 节“编译预处理”的内容更充实,划分更仔细,包括“文件包含处理”、“无参宏定义”、“带参宏定义”、“条件编译”。

6. 第 2 章还增加了“不同类型数据的混合运算”、“强制类型转换”、“混合赋值和复合的赋值运算符”、“逗号运算符”、“void 指针”、“转义字符”等内容。

7. 第 13 章“C++ 程序”中的“注释和输入输出语句”、“常量修饰符 const”、“模板函数”、“模板类”、“继承”、“派生类中的构造函数和析构函数”、“虚函数”、“纯虚函数和抽象类”等内容都做了改写。

8. 在主要章节的作业中增加了论述题(有关题目的基本内容可以在教材中找到),这有助于读者去发现 C、C++ 与数据结构的内在联系。而这种联系是由计算机科学本身的整体性决定的,通过自己独立的思考,认识这种整体性,有助于培养科学的态度,掌握正确的学习方法。

9. 配书多媒体软件的内容更丰富,与教材联系更紧密,使用更方便。

另外,我们对一些复杂算法,例如“迷宫求解”进行了改进,算法更简洁。

本书如果用作 C 和 C++ 教材,可选择第 1 章、第 2 章、第 3 章、第 4 章、第 7 章、第 13 章和第 14 章。从 C 语言“程序设计综合举例”中的典型算法过渡到数据结构中的“顺序表”和“新类型串”,再过渡到 C++。

如果用作数据结构教材,可选择第 3 章至第 12 章。如果想用 C++ 描述数据结构,我们在第 13 章中提供了将数据结构的 C 版本转化为数据结构 C++ 版本的算法实例。光盘文件夹 struct_vc 包含了数据结构 C++ 版本的典型程序。

本教材自 2002 年 3 月第 1 版发行后,两次印刷,在一年多的时间内又发行第 2 版,这要感谢天津师范大学各级领导和同仁的关怀和支持,感谢读者的认同与厚望。

读者有问题,欢迎直接与作者联系。

作者:王立柱

e-mail: data_structure@sohu.com

2003 年 7 月

目 录

C/C++ 与数据结构

第1章 机器语言程序简介	1
1.1 电子计算机的基本组成及其工作过程	1
1.2 机器语言程序设计	3
1.3 子程序调用过程	6
习题	8
第2章 C语言基础	9
2.1 基本类型变量和指针	9
2.2 基本类型	14
2.2.1 整型	14
2.2.2 字符型	15
2.2.3 实型	18
2.2.4 指针类型(地址类型)	18
2.2.5 不同类型数据的混合运算	18
2.2.6 强制类型转换	19
2.2.7 混合赋值和复合的赋值运算符	19
2.3 一维数组和指针	21
2.3.1 一维数组	21
2.3.2 指针	23
2.3.3 一维数组和指针变量的比较	25
2.3.4 void 指针	27
2.4 二维数组和指针	27
2.4.1 二维数组和二维指针	27
2.4.2 二维数组和一维指针	30
2.4.3 指针数组和指针的指针	31
2.5 结构和指针	34
2.5.1 结构变量	34

2.5.2	结构指针和数组	36
2.6	枚举类型	39
2.7	类型名选择	40
2.8	条件表达式与控制语句	41
2.8.1	关系运算	41
2.8.2	逻辑运算	42
2.8.3	条件控制语句	43
2.8.4	循环控制语句	48
2.8.5	break 语句和 continue 语句	52
2.8.6	逗号运算符	52
2.9	函数	53
2.9.1	函数与指针	53
2.9.2	函数调用与变量的存储类别	59
2.9.3	动态单元的分配和释放	66
2.10	指针参量	69
2.11	输入输出函数	72
2.11.1	printf 函数	72
2.11.2	scanf 函数	74
2.11.3	getchar 函数和 putchar 函数	77
2.12	编译预处理	78
2.12.1	文件包含处理	78
2.12.2	无参宏定义	80
2.12.3	带参宏定义	82
2.12.4	条件编译	84
2.13	流与文件	86
2.13.1	流与文件概述	86
2.13.2	文件打开关闭函数	87
2.13.3	无格式读写函数	89
2.13.4	字符读写函数	91
2.13.5	字符串读写函数	93
2.13.6	其他常用写函数	95
2.13.7	文件复制	97
2.14	初始化与赋值	98
2.15	程序设计综合举例	102
2.15.1	起泡排序	102
2.15.2	删除重复数据	104
2.15.3	数制转换	105
2.15.4	回文	105



2.15.5 约瑟夫问题(Josephus)	106
2.15.6 划分数组元素	108
2.15.7 模拟人工洗牌	109
2.15.8 三天打鱼两天晒网	110
2.15.9 筛法求素数	111
2.15.10 保序插入	112
2.15.11 折半查找	113
2.15.12 打印年历	113
习题	114
第3章 数据结构概论	117
3.1 数据结构和数据类型	117
3.2 算法和程序	120
3.3 算法分析	123
习题	126
第4章 顺序表	127
习题	134
第5章 顺序队列	135
5.1 顺序队列的定义	135
5.2 队列与文件	140
习题	141
第6章 顺序栈	142
6.1 顺序栈的定义	142
6.2 中缀表达式求值	146
习题	152
第7章 字符串	153
7.1 C语言串	153
7.2 新类型串	157
习题	167
第8章 链表	168
8.1 单向链式存储结构	168
8.2 单向链表	174
8.3 双向链式存储结构	178



8.4 双向链表	183
习题	187
第9章 链队列	188
9.1 链队列的定义	188
9.2 事件驱动模拟	190
习题	200
第10章 二叉树	201
10.1 二叉树的概念和性质	201
10.1.1 树和二叉树的概念	201
10.1.2 二叉树的性质	202
10.2 二叉树的存储	203
10.2.1 顺序存储结构	203
10.2.2 链式存储结构	205
10.3 二叉树层次遍历	206
10.3.1 层次遍历	206
10.3.2 建立二叉链表	208
10.3.3 垂直输出二叉树	209
10.4 二叉树的前序遍历	213
10.4.1 前序遍历	213
10.4.2 求二叉树从根至叶子的所有路径	215
10.4.3 求集合的幂集	218
10.4.4 快速排序	221
10.5 二叉树中序遍历	224
10.5.1 汉诺塔问题	224
10.5.2 由前序和中序序列建立二叉链表	226
10.6 二叉树后序遍历	228
10.7 递归评估	230
10.8 递归遍历的模拟	232
10.8.1 递归前序遍历的模拟	232
10.8.2 递归中序遍历的模拟	236
10.8.3 递归后序遍历的模拟	238
10.9 堆	241
10.10 哈夫曼树	248
10.10.1 哈夫曼树的定义	248
10.10.2 建立哈夫曼树	249
10.10.3 哈夫曼编码	251

10.11 二叉搜索树	252
10.12 平衡二叉搜索树	259
10.13 线索二叉树	266
习题	270
第 11 章 树	272
11.1 树的存储与遍历	272
11.1.1 树的存储	272
11.1.2 树的层次遍历	276
11.1.3 树的前序遍历	278
11.1.4 树的后序遍历	283
11.2 八皇后问题	285
11.3 八皇后解的图形输出	289
11.3.1 图形系统初始化	289
11.3.2 图形设计与实现	290
习题	298
第 12 章 图	299
12.1 图的概念和存储	299
12.1.1 图的概念	299
12.1.2 邻接矩阵表示法	300
12.1.3 邻接表表示法	304
12.2 图的遍历	305
12.2.1 广度遍历	305
12.2.2 深度遍历	307
12.3 最小生成树	309
12.4 单源最短路径	314
12.5 拓扑排序	319
12.6 关键路径	322
12.7 迷宫求解	326
12.8 骑士巡游和汉密尔顿路	332
习题	335
第 13 章 C++ 程序	336
13.1 C++ 对 C 的基本扩充	336
13.1.1 注释和输入输出语句	336
13.1.2 作用域说明	337
13.1.3 默认函数	338

13.1.4	引用	338
13.1.5	常量修饰符 const	340
13.1.6	内存的动态申请和释放	341
13.2	C++ 的基本要素	342
13.2.1	类和对象	342
13.2.2	this 指针	347
13.2.3	友元	348
13.2.4	外部运算符重载	350
13.2.5	初始化与赋值	351
13.3	重载	354
13.3.1	函数重载	354
13.3.2	成员和友元运算符重载	358
13.4	形式数据类型	362
13.4.1	模板函数	363
13.4.2	模板类	364
13.4.3	中缀表达式求值	367
13.5	继承和抽象类	369
13.5.1	继承	369
13.5.2	派生类中的构造函数和析构函数	373
13.5.3	虚函数	375
13.5.4	纯虚函数和抽象类	379
	习题	381
第 14 章	C++ 的 I/O 流库	382
14.1	屏幕输出	382
14.1.1	预定义的插入符	382
14.1.2	成员函数 put	384
14.1.3	成员函数 write	385
14.2	键盘输入	385
14.2.1	预定义的提取符	385
14.2.2	成员函数 get,getline	387
14.2.3	成员函数 read	389
14.3	插入符和提取符的重载	389
14.4	格式化输入输出	390
14.4.1	设置流的格式化标志	390
14.4.2	格式输出函数	392
14.4.3	操作子	393
14.5	文件	394

14.5.1	文件的打开关闭函数.....	394
14.5.2	无格式读写函数.....	396
14.5.3	字符读写函数.....	397
14.5.4	字符串读写函数.....	398
14.5.5	随机访问.....	400
14.5.6	文件错误处理.....	401
参考文献.....		403



第 1 章

机器语言程序简介

本章以机器语言程序的形式讲解对用任何程序语言编程都非常重要的知识:指令、地址、程序计数器、堆栈指示器、子程序调用过程、入口地址、断口地址和现场保护等。这是学习 C 语言和程序设计的基础。

1.1 电子计算机的基本组成及其工作过程

美国普林斯顿大学的冯·诺依曼于 1945 年提出计算机体系结构设计思想,一般称为“程序存储思想”,电子计算机从 1946 年问世至今都是以此为基本依据的,其中包括以下几个要点:

(1) 电子计算机应该采用二进制,而非十进制,因为二进制易于实现物理表示,其运算规则简单可靠。

(2) 操作指令也是一种信息,不妨用二进制表示。

(3) 程序和数据可以用完全相同的形式置于存储器中。

(4) 程序本身也可包含数据,即程序作为一组指令,每一条指令可分成操作码与操作数两部分,前者给出操作内容,后者给出数据所在的存储单元地址或直接给出数据。

比如,01 3000H 是一条操作指令。其中,01 是操作码,3000H 是操作数,表示将地址为 3000H 的单元中的数据放到寄存器 A(H 代表十六进制)。

冯·诺依曼型电子计算机的典型系统结构,如图 1.1 所示。

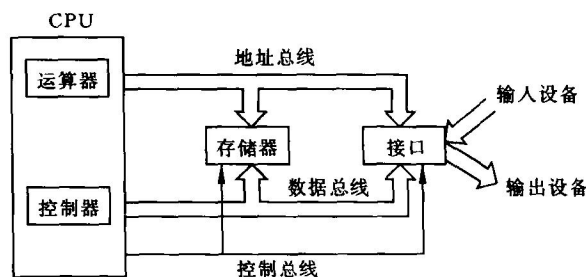


图 1.1 电子计算机系统结构框图

存储器用于存放程序指令及原始数据。运算器进行各种算术运算和逻辑运算。控制

器指挥和控制整个运算过程,使指令按要求一条一条地执行。利用输入设备输入指令代码和原始数据,输出设备显示或打印计算结果。

运算器和控制器合称为中央处理器,简称 CPU。

存储器由许多存储单元组成。在微型计算机中,每个存储单元通常可存放 8 位二进制信息,习惯上称为 1B。每个存储单元有一个编号,称为地址(码)。地址总线的多少决定了存储单元的多少,即决定了存储空间的大小。比如,16 根地址总线表明地址为 16 位二进制,存储器最多有 2^{16} B。地址 3000H 用二进制表示便为 0011 0000 0000 0000。

计算机的运行过程实质上是程序的执行过程,具体说:

首先编好一个程序,它包含一组指令,即一个指令序列。然后通过输入设备(比如键盘)将这组指令及所要处理的数据送入存储器。例如,表 1.1 是一个程序,计算 3 和 4 的和,它有 4 条指令。图 1.2 是这组指令和数据在存储器中的格式。

表 1.1 一个求和的程序

操作码	操作数	指令含义
01H	3000H	将地址为 3000H 的单元中的数据放入寄存器 A
03H	3001H	将地址为 3001H 的单元中的数据与寄存器 A 中的数据相加,结果存在 A
02H	3002H	将寄存器 A 中的数据存入地址为 3002H 的单元
07H		停机

执行程序时,控制器通过地址总线从存储器逐条取来指令执行。执行完第 1 条指令,寄存器 A 中的数据是 3;执行完第 2 条指令,A 中的数据是 7;执行第 3 条指令之后,地址 3002H 单元中的数据是 7;执行完第 4 条指令,结束。最后,输出设备将运算结果输出,比如在显示器上显示或通过打印机打印。

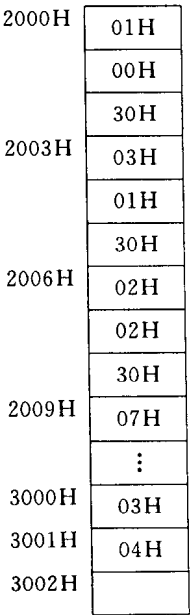


图 1.2 存储器中的程序

1.2 机器语言程序设计

与程序设计直接有关的是存储器,CPU 中的寄存器和指令系统。以微处理器 Z80 为例,如图 1.3 所示。

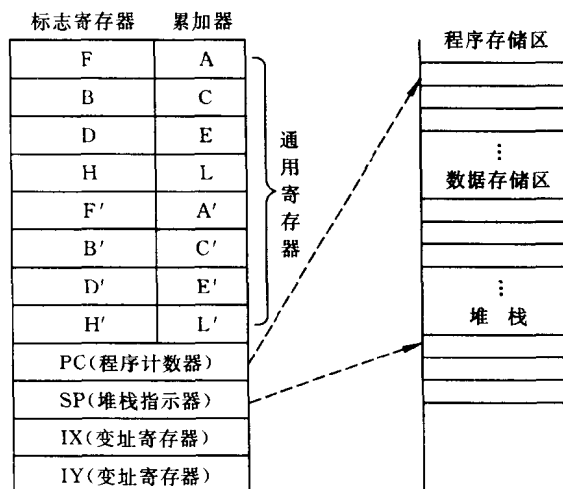


图 1.3 Z80 的编程模型图

、寄存器有两类：通用寄存器和专用寄存器。

B、C、D、E、H、L 代表 6 个 8 位通用寄存器,用来暂存操作数及运算的中间结果。它们也可以组合使用,比如 BC、DE、HL,作为 16 位通用寄存器。B'、C'、D'、E'、H'、L' 是它们的备用寄存器(或辅助寄存器)。

A、F、PC、SP、IX、IY 是专用寄存器。

A 是一个 8 位寄存器,通常称为累加器。它与算术逻辑运算器 ALU 一起完成各种运算。ALU 是一个组合逻辑电路,本身不能保留信息,只有与 A 寄存器一起才能完成各种运算;寄存器 A 在运算前向 ALU 提供操作数,运算后暂存运算结果。

F 是一个 8 位寄存器,一般称为标志寄存器。它与累加器 A 相连,记录运算结果的某些特征,以此作为控制程序流程转向的依据。

PC 为 16 位寄存器,习惯上称为程序计数器。程序是一组指令,这组指令一般都连续存放在存储器中。PC 用来寄存指令的地址。CPU 通过 PC 取来一条指令执行时,PC 便“指向”下一条指令,即 PC 的值变为下一条将要执行的指令的地址,而且,除非遇到转移指令或子程序调用指令外,CPU 都是通过 PC 顺序地提取一条一条的指令。比如,一条指令占 2 个字节,取出这条指令之后,PC 的值自动加 2。

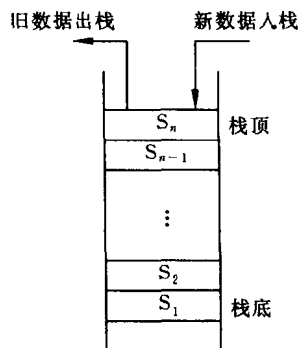


图 1.4 堆栈的结构示意图