



JISUANJI KEXUE JI BIANCHENG DAOLUN
计算机科学及编程导论

王立柱 王春枝 主编
Wang Lizhu Wang Chunzhi

欧阳勇 叶志伟 副主编
Ouyang Yong Ye Zhiwei



清华大学出版社



JISUANJI KEXUE JI BIANCHENG DAOLUN

计算机科学与编程导论

王立柱 王春枝 主编
Wang Lizhu Wang Chunzhi

欧阳勇 叶志伟 副主编
Ouyang Yong Ye Zhiwei

清华大学出版社

内 容 简 介

全书共 10 章,第 1 章~第 5 章介绍机器语言、C 语言和 C++ 语言的编程模式,第 6 章~第 8 章介绍递归、数据库和程序性能的基本内容,第 9 章介绍计算理论的基本概念,第 10 章介绍系统的组成和相关的概念及专业术语。全书内容翔实、连贯,结构清晰、严谨,目标具体、明确,为大专院校计算机专业程序设计课程的整合指出了一个方向。

本书适合作为计算机及相关专业本专科学生的教材,也可供计算机爱好者作为参考书。

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

计算机科学及编程导论/王立柱,王春枝主编. —北京:清华大学出版社,2015

ISBN 978-7-302-41135-2

I. ①计… II. ①王… ②王… III. ①电子计算机 IV. ①TP3

中国版本图书馆 CIP 数据核字(2015)第 183544 号

责任编辑:白立军

封面设计:傅瑞学

责任校对:胡伟民

责任印制:宋 林

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座

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

课件下载: <http://www.tup.com.cn>, 010-62795954

印 刷 者:三河市君旺印务有限公司

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185mm×230mm

印 张:12.5

字 数:310 千字

版 次:2015 年 9 月第 1 版

印 次:2015 年 9 月第 1 次印刷

印 数:1~2000

定 价:29.00 元

产品编号:065924-01

前 言

计算机科学以程序为基础,而且始于程序,也终于程序。本书是程序类课程学习的先导,也是基于程序的计算机专业知识学习的引领。

本书不是那种概念汇集、等量齐观的概括性模式,而是程序优先、程序生成概念、从低级到高级的实践——认知模式。

程序=算法+数据结构。数据结构是一组有组织的数据和对数据的基本操作,算法是用基本操作表示的对数据处理的有限步骤。数据结构一般都是用程序语言表示的,程序语言是数据结构描述的工具,因此程序也都是某种语言程序。算法越来越复杂,同时要求程序易写、易读、易维护,以及足够高效,这就带动了数据结构及程序语言的发展,发展的程序语言反过来又扩展了算法。算法和数据结构是一对矛盾,它们的相互作用推动了程序设计的发展。因为C和C++是核心程序语言,所以本书从中选择了部分有代表性的程序设计来说明这个发展过程,并以此为基础,引领计算机专业知识的学习。

本书内容概要

(1) 第1章机器语言模式。用若干条虚拟的简单机器指令作为虚拟的计算机,同时用一个简单的求和程序引入存储器、地址、寄存器、程序计数器等概念。然后用一个方程的两种求解程序,说明算法和指令集的相互作用关系,同时引入程序入口地址、子程序调用、断口地址、现场保护、栈、复杂指令集和精简指令集等概念。最后,通过分析机器语言的局限性,引入汇编语言、汇编程序等概念。

(2) 第2章C语言模式。为克服机器语言和汇编语言的局限性而进入基于简单类型的C语言模式,并用单个整数的处理程序说明这种语言的编程模式,然后用这个程序的变化来引入输入输出标准函数、循环结构、函数等概念。用一批整数的处理程序引入数组和指针、值传递和指针传递、const型指针、动态数组等概念。

(3) 第3章顺序表。数组是有局限性的,为克服它的局限性引入结构和顺序表。

(4) 第4章从C到C++。分析顺序表所暴露出来的C语言固有的局限性,为克服这些局限性引入内联函数、操作符函数、函数重载、引用型、函数模板等C++概念。

(5) 第5章顺序表类。运用第4章引入的C++概念,把C顺序表转换为C++顺序表类。这是一个从C到C++的平滑过渡模型,这曾是很多纠结于C和C++的人所心怀的希望。

(6) 第6章非线性结构与递归。通过斐波那契数列求解的非线性结构引入递归概念,然后通过汉诺塔、快速排序和八皇后问题,引入归纳证明法,它与递归紧密相关。

(7) 第7章数据库。通过C++的4个表处理程序引入关系数据库和结构化查询语言SQL。

(8) 第8章程序性能。通过典型程序的语句统计,引入空间复杂度和时间复杂度分析,其中涉及操作计数、步数、渐进记法等概念。

(9) 第9章计算理论。借助一个可以实现加1操作的图灵机,引入丘奇·图灵论题。借助一个通用语言Bare Bones,设计一个不可计算的函数,介绍P问题和NP问题,以及公钥密码学。

(10) 第10章计算机系统。沿着一个hello程序的生命周期,介绍计算机系统的组成和相关的概念,包括进程、线程、虚拟存储器、网络通信、高速缓存、层次存储结构等概念。

教学建议

教学章节	教学要求	课时 (含实验)
第1章 机器语言模式	熟悉例1.1和例1.2中的程序。 了解存储器、地址、寄存器、程序计数器、程序入口地址、子程序调用指令、断口地址、现场保护、栈、复杂指令集、精简指令集、汇编语言、汇编程序的概念	2
第2章 C语言模式	熟悉程序2.1~程序2.16,以及函数Josephus。 掌握循环结构。 掌握函数调用过程、函数原型、数组和指针的关系、值传递和指针传递、const型指针	8

续表

教学章节	教学要求	课时 (含实验)
第3章 顺序表	熟悉程序 3.1~程序 3.4。 熟悉顺序表声明和实现	4
第4章 从 C 到 C++	了解 C 语言本身的问题。 理解内联函数、操作符函数、函数重载、引用型、函数模板等 C++ 概念	6
第5章 顺序表类	熟悉从 C 顺序表到 C++ 顺序表类的转换过程。 掌握 new 和 delete 操作符。 熟悉类模板的概念	4
第6章 非线性结构与递归	熟悉汉诺塔、快速排序和八皇后算法。 掌握递归概念和归纳证明方法	4
第7章 数据库	熟悉程序 7.1~程序 7.4。 了解关系模型和查询语言 SQL 的基本语句	2
第8章 程序性能	了解操作计数、步数、渐进记法等概念	4
第9章 计算理论	熟悉“可以实现加 1 操作的图灵机”。 掌握 Bare Bones 通用语言。 了解丘奇·图灵论题、不可计算的函数、P 问题和 NP 问题以及公钥密码学	4
第10章 计算机系统	了解进程、线程、虚拟存储器、网络通信、高速缓存、层次存储结构等概念	2
总课时		40

致读者

与本书配套的教材有《C/C++ 与数据结构(第4版)》(上下册),王立柱主编,清华大学出版社出版。上册内容是 C 和 C++,下册内容是数据结构,它们是本书在程序设计方面的深入展开,本书把它们逻辑关联起来,帮助读者总体地把握。

读者如有问题,可以直接和作者联系。E-mail:tjwanglizhu@163.com。

编者

目 录

第 1 章 机器语言模式	1
1.1 模拟机器指令集与程序设计举例	1
1.2 机器语言的局限性	8
问题与练习	9
第 2 章 C 语言模式	10
2.1 基于基本类型的编程模式	11
2.2 对操作符的扩充——函数	17
2.3 函数原型和函数定义	21
2.4 代码抽象	23
2.5 对基本类型的扩充——数组和指针	23
2.5.1 数组	23
2.5.2 指针	25
2.5.3 数组和指针的深入讨论	28
2.5.4 选择排序	32
2.5.5 顺序搜索和二分搜索	35
2.5.6 值传递和指针传递	38
2.5.7 const 型指针	40
2.5.8 动态数组及其应用	43
问题与练习	47

第 3 章 顺序表	48
3.1 数组的局限性	48
3.2 结构体	50
3.3 顺序表的声明与实现	53
3.3.1 顺序表声明	53
3.3.2 顺序表的实现	57
3.4 数据抽象和封装	62
问题与练习	63
第 4 章 从 C 到 C++	64
4.1 C 语言固有局限性	64
4.2 内联函数	67
4.3 操作符重载和函数重载	68
4.3.1 操作符重载	69
4.3.2 函数重载	70
4.4 引用型	73
4.4.1 概念的由来	73
4.4.2 引用型及其应用	77
4.5 函数模板	79
4.6 提取符和插入符	81
4.7 默认参数	84
问题与练习	85
第 5 章 顺序表类	86
5.1 从 C 顺序表到 C++ 顺序表类	86
5.2 变量和对象	92
5.3 new 和 delete 操作符	94

5.4	需要增加的成员函数	95
5.5	顺序表类的声明和实现	99
5.6	类和类模板	102
5.7	基本类型的赋值形式扩展	107
	问题与练习	108
第 6 章	非线性结构与递归	109
6.1	树形结构与递归	109
6.2	C++ 递归函数	114
6.3	汉诺塔问题	116
6.4	快速排序	119
6.5	八皇后	122
	问题与练习	127
第 7 章	数据库	128
7.1	文件的读写操作	128
7.2	数据库的设计实例	130
7.3	关系数据库	139
	问题与练习	141
第 8 章	程序性能	142
8.1	什么是程序性能	142
8.2	空间复杂度	143
8.3	时间复杂度	145
8.3.1	操作计数	146
8.3.2	最好、最坏和平均操作计数	147
8.3.3	步数	148

8.3.4 渐进记法	151
问题与练习	152
第 9 章 计算理论	155
9.1 函数及其计算	155
9.2 图灵机原理和丘奇·图灵论题	156
9.3 通用程序设计语言——Bare Bones(基本要素)语言	159
9.4 一个不可计算的函数	162
9.5 迭代结构与递归结构的等价性	163
9.6 P 问题和 NP 问题	165
9.7 公钥密码学	168
9.7.1 模表示法	168
9.7.2 RSA 公钥密码系统	168
问题与练习	169
第 10 章 计算机系统	170
10.1 信息=位+上下文	170
10.2 编译系统	173
10.3 指令的读取和解释	174
10.3.1 系统的硬件组成	175
10.3.2 执行 hello 程序	177
10.4 操作系统	179
10.4.1 进程	181
10.4.2 线程	182
10.4.3 虚拟存储器	182
10.4.4 文件	183
10.5 网络通信	184

10.6 抽象	184
10.7 高速缓存	185
10.8 存储设备层次结构	186
问题与练习	187
 参考文献	 188

第 1 章 机器语言模式

指令集结构是对实际处理硬件的抽象。使用这个抽象,机器代码程序表现得就像它是运行在一个一次只执行一条指令的处理器上。——Randal E. Bryant

机器语言(machine language)是一个机器指令集,称为机器码(machine code),它们直接联系着计算机硬件,每一条机器指令都是计算机可以直接解读(即直接识别和执行)的数据。学好机器语言,需要深入了解计算机体系结构。但是这一章的重点不是机器语言程序设计,而是了解数据结构和算法这对矛盾最初是如何表现的,因此基本上不涉及计算机体系结构,只用若干条虚拟的机器指令集来模拟计算机。

1.1 模拟机器指令集与程序设计举例

在机器语言程序设计中,程序员亲自安排数据的存储,然后调用机器指令处理数据。

一条机器指令包含操作码和操作数两部分,前者是操作内容,后者是数据所在的存储单元地址,有时直接就是数据。例如,01H 3000H 是一条机器指令,其中 01H 是操作码,3000H 是操作数地址,具体意思是:取出地址为 3000H 存储单元中的数据,存入到 CPU 的寄存器 A 中。其中 H 表示操作码 01 和地址 3000 都是用十六进制表示的二进制数。1 位十六进制数和 4 位二进制数的对应关系如表 1.1 所示。

表 1.1 1 位十六进制数和 4 位二进制数的对应表

十六进制	二进制	十六进制	二进制	十六进	二进制	十六进制	二进制
0	0000	2	0010	4	0100	6	0110
1	0001	3	0011	5	0101	7	0111

续表

十六进制	二进制	十六进制	二进制	十六进	二进制	十六进制	二进制
8	1000	A	1010	C	1100	E	1110
9	1001	B	1011	D	1101	F	1111

计算机存储器是一系列连续的字节,每一个字节有 8 位。指令 01H 3000H 需要 3 个字节。

表 1.2 是一台模拟计算机的机器指令集。

表 1.2 一台模型计算机的机器指令集

指令名称	机器指		指令说明
	操作码	操作数	
取数	01H	N	$A \leftarrow (N)$: 从地址为 N 的单元读取数据,存入寄存器 A
直接取	10H	C	$A \leftarrow C$ 把数据 C 存入寄存器 A
存数	02H	N	$(N) \leftarrow A$: 从寄存器 A 中读取数据,存入地址为 N 的单元
加法	03H	N	$A \leftarrow A + (N)$: 将地址为 N 的单元中的数据经过某一个寄存器和寄存器 A 中的数据相加,结果存入 A
直接加	30H	C	$A \leftarrow A + C$ 将数据 C 经过某一个寄存器和寄存器 A 中的数据相加,结果存入 A
乘法	04H 06H	N N	$A \leftarrow A \times (N)$: 将地址为 N 的单元中的数据经过某一个寄存器和寄存器 A 中的数据相乘,结果存入 A
转移	00H		$PC \leftarrow N$: 用地址 N 更新程序计数器 PC 的值
停机			停机

例 1.1 编程计算 $z = x + y$ 。

算法步骤:

- (1) 从 x 中读数。
- (2) 从 y 中读数,计算 $x + y$ 。
- (3) 把求和结果存到 z 。

首先把 x 、 y 和 z 安排在存储器数据区,地址依次为 3000H、3001H 和 3002H,如

图 1.1(a)所示。

算法的步骤可以用表 1.2 中的机器指令表示,其对应关系如表 1.3 所示。

表 1.3 求和算法和程序

算法步骤	机器指令	指令说明
从 x 中读数	01H 3000H	取出地址为 3000H 单元中的数据,存入寄存器 A
从 y 中读数,计算 $x+y$	03H 3001H	将地址为 3001H 单元中的数据与寄存器 A 中的数据相加,结果存在寄存器 A
把求和结果存到	02H 3002H	将寄存器 A 中的数据存入地址为 3002H 的单元
	00H	停机

机器指令位于存储器的程序区,除停机指令之外,其他指令都占 3 个字节,机器指令首字节地址称为指令地址。与算法步骤对应的 4 条机器指令的地址依次是 2000H、2003H、2006H 和 2009H,如图 1.1 所示。CPU 中的程序计数器 PC 存储机器指令地址,CPU 不断从 PC 中提取指令执行,直到停机为止。每执行一条指令,PC 都会指向下一条指令。例 1.1 的执行过程如图 1.1 所示。

例 1.2 编程计算 $y=2x^2+3x+4$ 。

算法设计:

将原方程分解为 $y=(2x+3)x+4$,据此得到算法步骤如下:

- (1) 取 2。
- (2) 从 x 中读数,计算 $2 \times x$ 。
- (3) 取 3,计算 $2x+3$ 。
- (4) 从 x 中读数,计算 $(2x+3) \times x$ 。
- (5) 取 4,计算 $(2x+3)x+4$ 。
- (6) 将计算结果存入 y 。

将 x 和 y 安排在存储器的数据区,地址分别为 3003H 和 3004H。

每一步骤可以用表 1.2 中的机器指令来表示,其对应如表 1.4 所示。

例 1.2 的算法还可以用如下步骤描述。

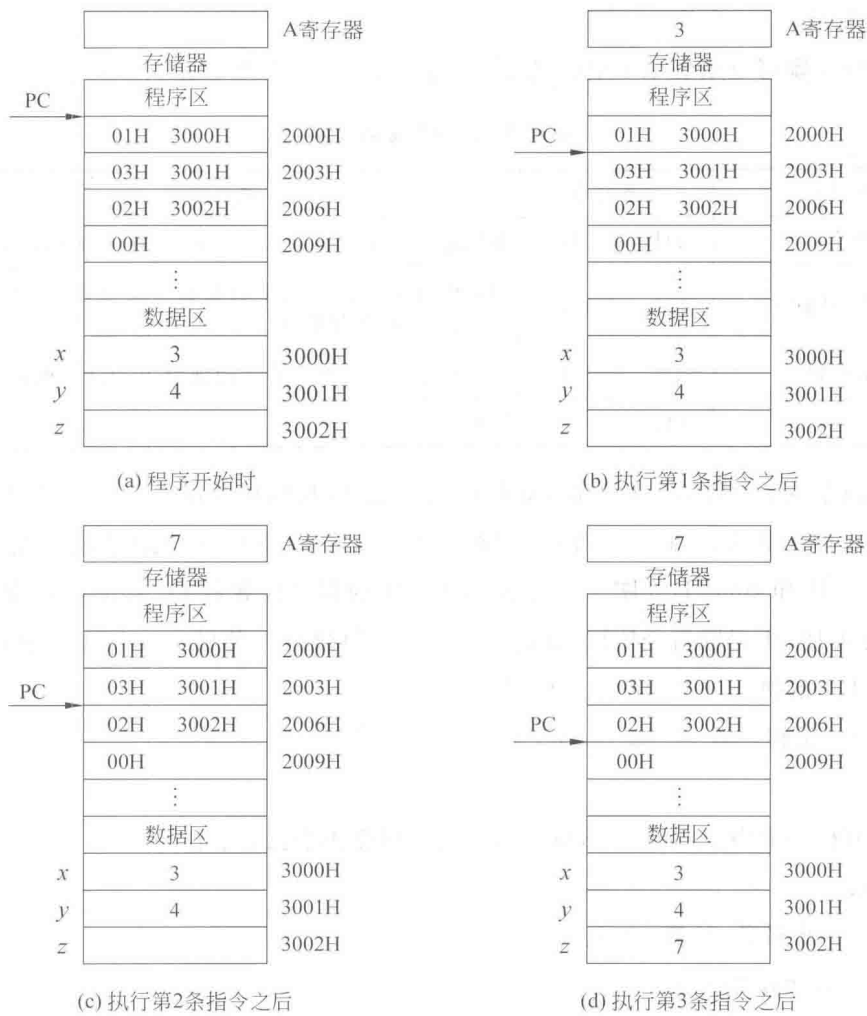


图 1.1 求和程序 $z=x+y$ 的执行过程示意图

- (1) 取 2。
- (2) 从 x 中读数, 计算 $2x$ 。
- (3) 从 x 中读数, 计算 $2xx$ 。
- (4) 取 3。

(5) 从 x 中读数, 计算 $3x$ 。

(6) 计算 $2xx+3x$ 。

表 1.4 例 1.2 的算法和程序

算法步骤	机器指令	指令说明
取 2	10H 2	$A \leftarrow 2$
从 x 中读数, 计算 $2x$	04H 3003H	$A \leftarrow A * (3003H)$
取 3, 计算 $2x+3$	30H 3	$A \leftarrow A + 3$
从 x 中读数, 计算 $(2x+3)x$	04H 3003H	$A \leftarrow A \times (3003H)$
取 4, 计算 $(2x+3)x+4$	30H 4	$A \leftarrow A + (3002H)$
将计算结果存入 y	30H 3004H	$A \leftarrow A + 4$
	00H	停机

(7) 取 4, 计算 $2xx+3x+4$ 。

但是, 利用 1.2 的指令集, 步骤(4)是无法直接实现的。因为在执行步骤(4)之前, 寄存器 A 存储着 $2xx$ 的计算结果, 这个结果以后还有用, 而步骤(4)需要寄存器来存储 3, 这与寄存器发生了冲突。解决的办法是, 设计一个新的变量, 例如 z , 在执行步骤(4)之前, 用操作码为 02H 的指令把寄存器中 $2xx$ 的计算结果保存到 z , 然后执行步骤(4)和(5), 接下来, 在步骤(6)中, 利用操作码为 03H 的指令, 读取 z 中的值与寄存器中的值相加。

从算法的实现来讲, 这涉及计算的中间结果保护和 CPU 的资源共享问题, 这是一个复杂的问题, 是一个程序员无法驾驭的; 而从算法的设计来讲, 这又是一个化复杂为简单的普遍方法。于是产生了一个尖锐的矛盾: 机器指令限制了算法设计的正常发展。解决这个矛盾的方法是增加相关的机器指令, 如表 1.5 所示。

表 1.5 与子程序调用有关的指令

指令名称	机器指令		指令说明
	操作码	操作数	
子程序调用	07H	N	断口地址进栈, 用地址 N 更新程序计数器 PC 的值

续表

指令名称	机器指令		指令说明
	操作码	操作数	
返回主程序	08H		断口地址出栈,用来更新程序计数器 PC 的值
压栈	09H	T	寄存器 T 中的数据进栈保存
出栈	0aH	T	出栈,出栈数据进入寄存器 T

下面结合表 1.5,介绍一些概念。

入口地址。程序是一组指令,第一条指令的地址称为程序的入口地址。例如,例 1.1 中的 2000H。

中断和断口地址。一个程序在执行过程中去执行另一个程序,称为中断,前者称为主程序,后者称为子程序。主程序中断时的下一条指令的地址称为主程序的断口地址。

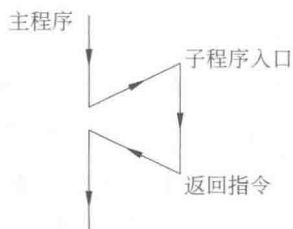


图 1.2 子程序的调用过程

现场保护和现场恢复。在主程序中中断时,如果有些寄存器还存放着主程序的中间结果,那么就要将中间结果放入异地暂存,把寄存器“让给”子程序,这个过程是现场保护。待子程序执行完毕,再将主程序的中间结果放回寄存器,这个过程是现场恢复。

子程序调用过程。当主程序转去执行子程序时,首先将断口地址压入栈保存,再将子程序的入口地址送入 PC。进入子程序后,首先保护主程序现场,然后执行子程序,子程序执行完毕后,恢复主程序现场,然后将断口地址从栈顶送回 PC,过程如图 1.2 所示。

例 1.3 在增加了子程序调用指令的条件下,重新编程计算 $y=2x^2+3x+4$ 。

首先将方程分解为两个简单的方程:

$$(1) y=2x^2+3x+4。$$