·高等学校计算机基础教育教材精选·

C语言程序设计

马希荣 熊聪聪 刘文菊 王立柱 编著

清华大学出版社

·高等学校计算机基础教育教材精选·

C语言程序设计

马希荣 熊聪聪 刘文菊 王立柱 编著

清华大学出版社
北京

内 容 简 介

指针和函数是C语言的核心内容，也是学习的难点，本书就是从系统地解决这两个难点的目标出发来精炼教材结构和内容的。全书共13章，主要内容包括：基本数据类型、表达式与操作符、程序流程控制、一级指针和一维数组、函数、模块化程序设计、字符串、结构体、文件、C综合程序设计、二维数组和指针、高级程序设计。

本书提供了丰富的综合程序设计例题和高级程序设计案例，以帮助学习者提高综合运用C语言知识进行程序设计的能力。同时，本书注重程序语言的发展规律，在内容组织和讲解方面力求实现“从C平滑过渡到C++”的思想。

本书配套光盘提供了精心制作的多媒体教学软件，实现了三级菜单与教材章节一一对应，算法、程序代码、抽象结构、存储结构、运行过程和结果同时展现，蓝色光条跟踪程序执行结果。

本书适用于高校本科各专业C语言程序设计课程，教师和学生可以根据各自的专业特点选择相关的内容。本书也可供程序设计初学者自学参考。

图书在版编目（CIP）数据

C语言程序设计 / 马希荣等编著. —北京：清华大学出版社，2010.2
ISBN 978-7-302-21924-8

Ⅰ. ①C… Ⅱ. ①马… 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(2010)第016289号

责任编辑：战晓雷
责任校对：白　蕾
责任印制：孟凡玉

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：北京富博印刷有限公司
装 订 者：北京市密云县京文制本装订厂
经　　销：全国新华书店
开　　本：185×260　　**印　张**：14.25　　**字　数**：321千字
（附光盘一张）
版　　次：2010年2月第1版　　**印　次**：2010年2月第1次印刷
印　　数：1～4000
定　　价：29.00元

出版者的话
PUBLISHER'S NOTES

为了使教材建设进入良性循环，从而实际地提高教材质量，我们在进行一项新的教材编写方法的尝试：由已有的优秀教材作者和教材的应用院校的教师合作，更有针对性地开发新教材，既保留原教材的创新点和特点，又融进一些应用院校的具体需要和建议，既夯实良好的基础，又不断做切实的改进。

本书是“十一五”国家级规划教材《C/C++与数据结构》(第3版)作者、教育部—微软精品课主持人王立柱教授，和天津师范大学计算机与信息工程学院马希荣教授、天津科技大学计算机科学与信息工程学院熊聪聪教授、天津工业大学计算机科学与软件学院刘文菊教授共同编写的。根据相关院校的课程设计特点，提取了原教材的C语言部分，同时在章节安排上作了调整，并根据相关院校的学生特点，对一些难点和重点知识做了更细致和透彻的讲解。

原教材在指针和函数这两个重点内容的表述方法上有独到之处，多媒体课件也别具一格。本教材在保留和发扬这些特色的同时，实例更丰富，内容更紧凑，难易搭配更合理，适用面更广。

本书属于教育部—微软精品课“数据结构”建设项目的系列教材之一。

前言 FOREWORD

指针和函数是C语言的核心内容，也是学习的难点，本书就是从系统地解决这两个难点的目标出发来精炼教材内容及铸造教材结构的。

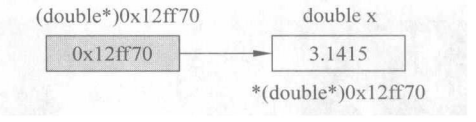
例如，本书以第1章“机器语言程序简介”为引入点，借助指针字面值常量（如下图所示）的概念，从多个角度来阐述和运用指针：

① 指针是类型化的地址；地址是直接引用的指针。

② 一个变量等价于一个长度为1的一维数组。一维数组是一组类型相同、空间相邻的变量。

③ 一个 m 行 n 列的二维数组等价于一个长度为 $m * n$ 的一维数组。一个长度为 n 的一维数组等价于一个1行 n 列的二维数组。

④ 二维数组是元素为一维指针常量的一维数组，每一个指针常量分别指向长度相同、空间相邻的一维数组。指针数组是元素为一维指针变量的一维数组，每一个指针变量可以分别指向长度不同且空间不相邻的一维数组。



又如，本书以变量的初始化为前提，严格表示函数调用的三个步骤（举例见下表）：

程序 6.1 数组求和	调用的内部执行过程

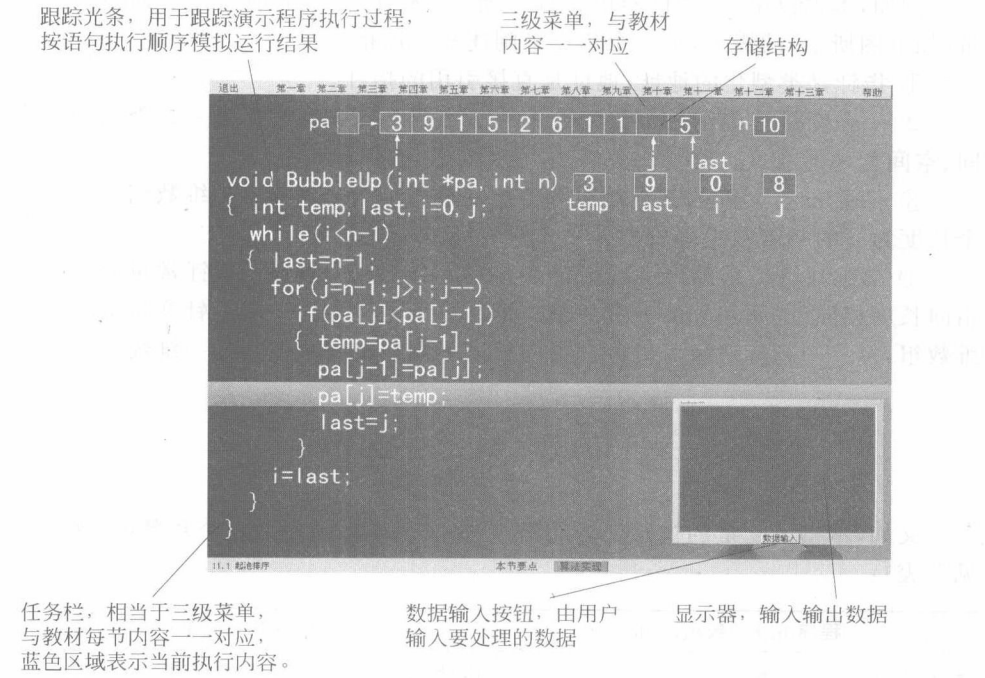
`#include<stdio.h>`
`int Sum(int*pa,int n)`
`//①int*pa=a; int n=6;`
`{`
`int i,t=0;`
`for(i=0;i<n;i++)  //累计求和`
`t=t+pa[i];`
`return(t);  //②int _temp=t;`
`}`
`int main(void)`
`{`
`int a[6]={10,11,12,13,14,15},s;`
`s=Sum(a,6);`
`//Sum(a,6); ③s=_temp;`
`printf("sum=%d\n",s);`
`return(0);`
`}`
	假设_temp是系统建立的临时变量，它存储返回值，调用函数从中取值。临时变量在主函数调用语句结束时被销毁。
	`int a[6]={10,11,12,13,14,15},s;`
	`int *pa=a; int n=6;  //Sum(a,6);`
	`int i,t=0;`
	`for(i=0;i<n;i++)  //累计求和`
	`t=t+pa[i];`
	`int _temp=t;  //return(t);`
	`s=_temp;  //s=Sum(a,6);`
	`printf("sum=%d\n",s);`

① 主调函数通过实参给被调函数的形参初始化。

② 如果被调函数有返回值，那么系统根据函数返回值类型创建一个临时变量(假设名为_temp)，并通过初始化将返回值存储到这个临时变量。

③ 主调函数如果需要被调函数的返回值，就从这个临时变量取值。

本书配备独具特色的多媒体教学软件。教学软件的三级菜单与教材三级目录一一对应，直观展示程序运行每一步骤和环节，既可助教又可助学，有助于解决学生能力参差不齐与课程内容多、课时不足、算法复杂抽象、难以理解的矛盾；模拟程序调试过程，弥补了实验课时的不足。既便于学生自学，又为教师搭建了可以充分展示自己的平台(如图所示)。



本书配有丰富的程序设计实例，特别是综合设计实例，使学生可以综合运用所学的知识，提高综合设计能力。

本教材配有实验和习题解答同时出版。

读者如有问题或发现本书及配套教学软件中的错误，欢迎直接与作者联系，作者将不胜感激。

E-mail：tjwanglizhu@163.com

精品课网址：http://59.67.71.237:8080/dsc

作　者

2010年1月

目录 CONTENTS

第1章

机器语言程序简介

C语言的核心部分是指针和函数，也是学习的难点，而简单了解一些硬件部分即机器语言程序的内容，有助于我们认识和理解这两个知识点。

1.1 计算机组成及工作过程

美国普林斯顿大学的冯·诺依曼于1945年提出的计算机体系结构设计思想，一般称为“**程序存储思想**”。计算机从1946年问世至今都是以这种思想为基本依据的，这个思想主要包含如下4个内容：

① 计算机应该采用二进制，与十进制相比，实现二进制运算的结构简单，容易控制。

② 操作指令也是一种信息，可以用二进制代码表示。

③ 程序和数据的存储形式可以完全相同。

④ 程序中的每一条指令由操作码和操作数两部分组成，前者是操作内容，后者是数据所在的存储单元地址或直接就是数据。

例如，“01H 1000H”是一条操作指令，其中01H是操作码，1000H是操作数。具体内容是“将地址为1000H存储单元中的数据放到CPU中的寄存器A中”。

冯·诺依曼型计算机的典型系统结构如图1.1所示。

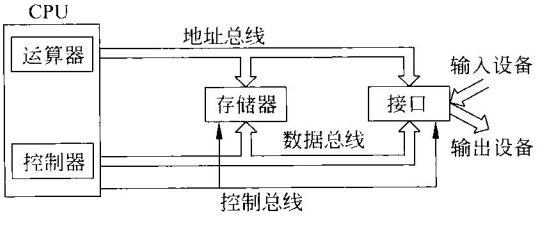


图1.1 电子计算机系统结构框图

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**控制和指挥整个运算过程，使指令按要求一条一条执行。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**。CPU通过数据总线与存储器和接口交换信息。

存储器由存储单元组成，每个存储单元习惯上称为一个**字节**(1B)。一个字节有8位，存

放 8 位二进制信息。每个字节有一个编号，称为**地址**(码)。

存储器的大小是指它有多少字节。每个字节相当于一座房子，有多少座房子，就应该有多少地址；反之，有多少地址，就有多少座房子。地址总线的多少决定了地址的多少，也就是字节的多少。设有 16 根地址总线，如果每根地址线有脉冲信号表示 1，没有脉冲信号表示 0，那么 16 根地址总线相当于 16 位二进制的地址表达方法，可以有 2^{16} 个地址，因此存储器最多有 2^{16}B。

例如，1000000000000000 表示第 1 根地址线有脉冲信号，其他地址线没有脉冲信号，它代表一个二进制地址。这个地址可以用一个 4 位十六进制数简单地表示为 8000H (H 代表十六进制)。4 位二进制共有 16 个值，和 1 位十六进制数的 16 个值一一对应，见表 1.1。这个一一对应关系可以将一个 16 位二进制数简单地表示为一个 4 位十六进制数。例如 0011000000001111 对应 300fH。使用十六进制形式仅仅为了简洁。类似，如果有 32 根地址线，那么 32 位二进制地址可以用 8 位十六进制数简洁表示。

表 1.1 4 位二进制数和 1 位十六进制数对应表

二进制	十六进制	二进制	十六进制	二进制	十六进制	二进制	十六进制
0000	0	0100	4	1000	8	1100	c
0001	1	0101	5	1001	9	1101	d
0010	2	0110	6	1010	a	1110	e
0011	3	0111	7	1011	b	1111	f

存储器分为程序存储区、数据存储区和栈。程序存储区存放程序中的指令。数据存储区存放数据，如图 1.2 所示。CPU 通过地址总线发出相应的地址，选中存储器的某一字节即存储单元，然后通过数据总线操作该单元中的数据。

CPU 中设有寄存器，与运算器或控制器直接相连，可以存放数据或计算的中间结果，如图 1.2 所示。因为不通过地址总线和数据总线，所以寄存器的数据存取速度快。但是寄存器不能无限制增加，多了就会影响速度。为了解决这个矛盾，我们在存储器中特别划分出一块区域，称为**栈**，其特点是存取数据都在一端，该端称为**栈顶**，如图 1.3 所示。栈的存取不需要计算地址，因此速度快，可以作为寄存器的补充。

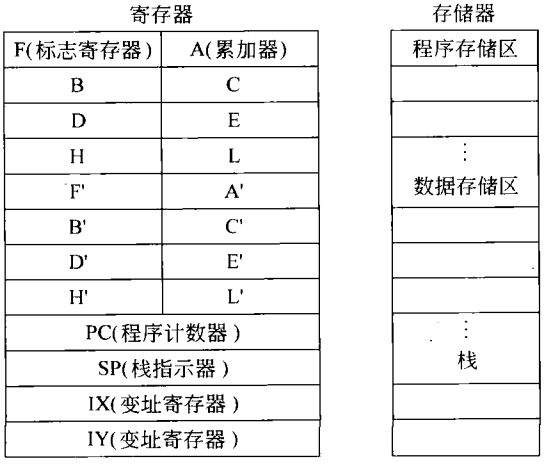


图 1.2 Z80 编程模型图

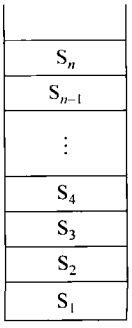


图 1.3 栈的结构示意图

CPU 中的寄存器有两类：通用寄存器和专用寄存器。A、B、C、D、E、H、L 是 8 位通用寄存器，是运算器的组成部分，用来暂存操作数及运算的中间结果。它们可以组合使用，即 BC、DE、HL，作为 16 位通用寄存器。A'、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：累加器 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。除非遇到转移指令或子程序调用指令，CPU 都是通过 PC 顺序地提取指令。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。

SP 为 16 位寄存器，习惯上称为**堆栈指示器**。SP 的值始终是栈顶元素的地址，随着数据的存入和删除，SP 的值自动改变。

IX 和 IY 是两个独立的 16 位变址寄存器，通常包含一个基地址（这个地址是根据需要写入的，一般在程序的首部通过赋值完成），由基地址加上偏移量（在程序运行中给出），以形成操作数的实际地址。

程序是一组指令，指令联系着存储器和 CPU。下面我们通过一个程序，了解计算机组成原理和工作过程。这是一个简单的求和程序：

```
y=3+4
```

3、4 和 y 存储在存储器中的数据存储区，地址依次为 3000H、3001H 和 3002H。如图 1.4(a)所示。注意，y 不是数据，只是一个存储单元的标识符，这个单元将要存放计算结果。程序由 4 条指令组成，如表 1.2 所示，这组指令依次存储在存储器中的程序存储区，地址分别为 2000H、2003H、2006H 和 2009H，前 3 条指令各占 3 个字节，第 4 条指令占一个字节。一条指令(013000H)的实际存储如图 1.5 所示。

表 1.2　求和程序 y=3+4 所包含的指令

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，结果留在 A
02H	3002H	将寄存器 A 中的数据存入地址为 3002H 的单元
00H		停机

CPU 从程序计数器 PC 依次提取指令执行，每条指令的意义如表 1.2 所示。图 1.4(a)～图 1.4(d)演示了指令执行的结果，第 4 条指令执行之后程序停止。

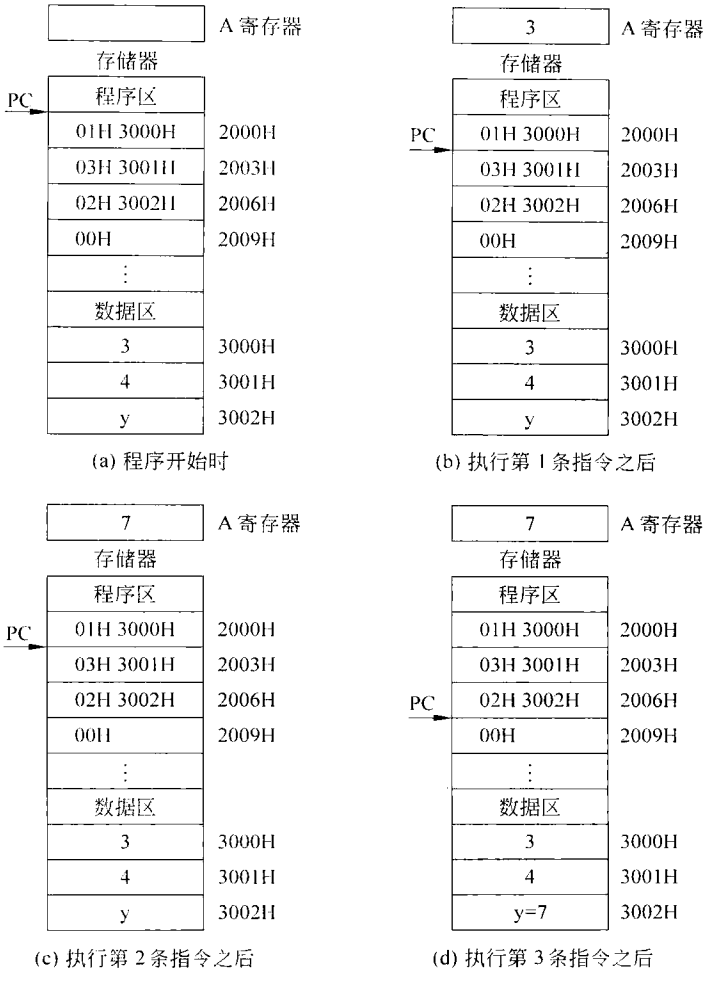


图 1.4 求和程序 y=3+4 的执行过程示意图

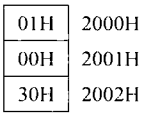


图 1.5 指令 013000

1.2 机器语言程序设计

对计算机而言，它的各个硬件部分存在的意义和相互的联系通过指令来体现。对程序设计来说，计算机就是指令系统。为了设计程序，我们假设一台模型计算机具有几条简化的指令，见表 1.3。

例 1.1 编程计算 $y=ax^2+bx+c$。

算法设计：

① 设计算法：函数分解为 $y=(ax+b)x+c$。按四则运算规则，算法步骤见表 1.4 第 1 列。

② 分配存储单元：数据的存储和与算法步骤一一对应的指令的存储见表 1.4 第 2 列和第 3 列。

表 1.3　一台模型计算机的指令系统

指令名称	机器指令		说　　明
	操作码	操作数	
取数	01H	N	A←(N)将地址为 N 的单元的数据存入寄存器 A
存数	02H	N	(N)←A 将寄存器 A 的数据存入地址为 N 的单元
加法	03H	N	A←A+(N)将地址为 N 的单元的数据和 A 中的相加，结果存入 A
乘法	04H	N	A←A×(N)将地址为 N 的单元的数据和 A 中的相乘，结果存入 A
停机	00H		停机

表 1.4　例 1.1 程序清单

算　　法	存　储　器	指令存储地址	说　　明
	程序存储区		
取数 a	01H 3000H	2000H	A←(3000H)
计算 $a\times x$	04H 3003H	2003H	A←A*(3003H)
计算 $a\times x+b$	03H 3001H	2006H	A←A+(3001H)
计算 $(a\times x+b)\times x$	04H 3003H	2009H	A←A×(3003H)
计算 $(a\times x+b)\times x+c$	03H 3002H	200cH	A←A+(3002H)
计算结果存入 y	02H 3004H	200fH	(3004H)←A
算法结束	00H	2012H	停机
	数据存储区	数据存储地址	
	a	3000H	
	b	3001H	
	c	3002H	
	x	3003H	
	y	3004H	

能否采用另外一种算法：

① 取数 a

② 计算 $a\times x$

③ 计算 $a\times x\times x$

④ 计算 $a\times x\times x+b\times x+\cdots$

根据目前的指令系统，步骤④不能对应一条指令，因此目前不能采用。

1.3　子程序调用过程

例 1.2　在例 1.1 编程计算 $y=ax^2+bx+c$ 的时候，因为指令系统的限制而没有采用一种适合我们习惯的算法，现在细化这个算法步骤：

① 取数 a

② 计算 $a\times x$

③ 计算 $a\times x\times x$

④ 计算 $b\times x$

⑤ 计算 $a\times x\times x+b\times x$