

21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

C语言 程序设计

C Programming

朱立华 王立柱 编著

- 高屋建瓴 透彻解析指针概念
- 循序渐进 交待函数来龙去脉
- 简明扼要 推演和实验同步
- 学以致用 应试与能力并重
- 综合直观 全方位多媒体演示



精品系列

 人民邮电出版社
POSTS & TELECOM PRESS

清华大学出版社“十二五”规划教材

C语言 程序设计

第2版

陈国君 主编

- 1. 全面覆盖C语言知识点
- 2. 注重C语言编程能力的培养
- 3. 注重C语言与C++语言的联系
- 4. 注重C语言在工程中的应用
- 5. 注重C语言在嵌入式系统中的应用



清华大学出版社

清华大学出版社
Tsinghua University Press

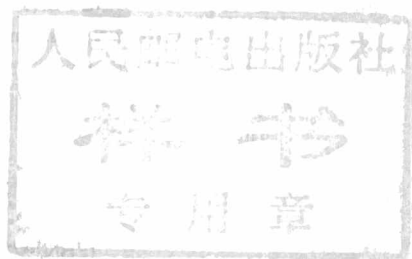
21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

C语言 程序设计

C Programming

朱立华 王立柱 编著



精品系列

人民邮电出版社

北京

图书在版编目 (CIP) 数据

C语言程序设计 / 朱立华, 王立柱编著. —北京: 人民邮电出版社, 2009. 9
21世纪高等学校计算机规划教材
ISBN 978-7-115-19957-7

I. C… II. ①朱…②王… III. C语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆CIP数据核字 (2009) 第137312号

内 容 提 要

本书是为以C语言作为入门语言的程序设计课程而编写的, 以培养学生具有基本的程序设计能力为主要目标。

全书共分13章, 内容包括: C语言的基本概念、常量与变量、运算符及表达式、顺序结构、分支结构、循环结构、数组、函数、指针、结构体、联合体和枚举类型、预处理和标准函数、文件等。同时, 还介绍了程序设计的基本方法和主要算法, 并给出了一个综合的小型管理系统的设计与实现, 初步引入了C++语言的基本知识。

本书对C语言最精华的知识——指针和函数作了全面透彻的讲解和分析, 所选例题有很强的针对性, 并对例题的源代码和运行结果有详细的注释。各章附有适量的课后习题, 便于学生复习巩固。

本书配有两个版本的免费多媒体课件, 程序的动态演示使很多初学者难以理解的问题变得直观、简单、易懂, 既方便了教师教学, 又方便了学生自学。

另外, 本书还配有辅导教材《C语言程序设计习题解析与实验指导》, 对主教材后的习题作详细解析, 并配有大量补充习题, 同时还精心设计了10个实验。

本书适合作为C语言程序设计课程的教材, 也可以作为广大计算机编程爱好者学习C语言的自学教材和参考书。

21世纪高等学校计算机规划教材

C语言程序设计

-
- ◆ 编 著 朱立华 王立柱
责任编辑 滑 玉
执行编辑 武恩玉
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京艺辉印刷有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 19.5
字数: 509千字
印数: 1—4000册
- 2009年9月第1版
2009年9月北京第1次印刷

ISBN 978-7-115-19957-7

定价: 32.00元

读者服务热线: (010) 67170985 印装质量热线: (010) 67129223

反盗版热线: (010) 67171154

出版者的话

计算机应用能力已经成为社会各行业从业人员最重要的工作要求之一，而计算机教材质量的好坏会直接影响人才素质的培养。目前，计算机教材出版市场百花争艳，品种急剧增多，要从林林总总的教材中挑选一本适合课程设置要求、满足教学实际需要的教材，难度越来越大。

人民邮电出版社作为一家以计算机、通信、电子信息类图书与教材出版为主的科技教育类出版社，在计算机教材领域已经出版了多套计算机系列教材。在各套系列教材中涌现出了一批被广大一线授课教师选用、深受广大师生好评的优秀教材。老师们希望我社能有更多的优秀教材集中地呈现在老师和读者面前，为此我社组织了这套“21世纪高等学校计算机规划教材——精品系列”。

本套教材具有下列特点。

(1) 前期调研充分，适合实际教学需要。本套教材主要面向普通本科院校的学生编写，在内容深度、系统结构、案例选择、编写方法等方面进行了深入细致的调研，目的是在教材编写之前充分了解实际教学的需要。

(2) 编写目标明确，读者对象针对性强。每一本教材在编写之前都明确了该教材的读者对象和适用范围，即明确面向的读者是计算机专业、非计算机理工类专业还是文科类专业的学生，尽量符合目前普通高等教育计算机课程的教学计划、教学大纲以及发展趋势。

(3) 精选作者，保证质量。本套教材的作者，既有来自院校的一线授课老师，也有来自IT企业、科研机构等单位的资深技术人员。通过他们的合作使老师丰富的实际教学经验与技术人员丰富的实践工程经验相融合，为广大师生编写出适合目前教学实际需求、满足学校新时期人才培养模式的高质量教材。

(4) 一纲多本，适应面宽。在本套教材中，我们根据目前教学的实际情况，做到“一纲多本”，即根据院校已学课程和后续课程的不同开设情况，为同一科目提供不同类型的教材。

(5) 突出能力培养，适应人才市场要求。本套教材贴近市场对于计算机人才的能力要求，注重理论知识与实际应用的结合，注重实际操作和实践动手能力的培养，为学生快速适应企业实际需求做好准备。

(6) 配套服务完善。对于每一本教材，我们在教材出版的同时，都将提供完备的PPT课件，并根据需要提供书中的源程序代码、习题答案、教学大纲等内容，部分教材还将在作者的配合下，提供疑难解答、教学交流等服务。

在本套教材的策划组织过程中，我们获得了来自清华大学、北京大学、人民大学、浙江大学、吉林大学、武汉大学、哈尔滨工业大学、东南大学、四川大学、上海交通大学、西安交通大学、电子科技大学、西安电子科技大学、北京邮电大学、北京林业大学等院校老师的大力支持和帮助，同时获得了来自信息产业部电信研究院、联想、华为、中兴、同方、爱立信、摩托罗拉等企业和科研单位的领导和技术人员的积极配合。在此，向他们表示衷心的感谢。

我们相信，“21世纪高等学校计算机规划教材—精品系列”一定能够为我国高等院校计算机教学做出应有的贡献。同时，对于工作欠缺和不妥之处，欢迎老师和读者提出宝贵的意见和建议。

表 1

(书中的表 11.3) 数值型数据的终端(文件)输入/输出

终端输入/输出	文件输入/输出
<pre>int n; scanf("%d",&n); //从键盘读入一个整数 printf("n=%d\n",n); //在显示器上输出一个整数</pre>	<pre>int n; fscanf(stdin,"%d",&n); fprintf(stdout, "n=%d\n",n);</pre>

表 2

(书中的表 6.6) 判断质数程序的两种形式对比

程序 6.4 判断质数的函数调用形式	(程序 4.7) 判断质数的主函数形式
<pre>#include<stdio.h> #include<math.h> int primeInt(int m); //判断质数函数声明 int main() { int m; scanf("%d",&m); if (primeInt(m)) //判断质数函数调用 printf("%d is a prime\n",m); else printf("%d is not a prime\n",m); return 0; } int primeInt(int m) //判断质数函数定义 { int i,k,prime=1; //函数内定义 3 个变量 if (m==1) //形参若为 1 prime=0; //形参若为 0, 非质数 k=(int)sqrt(m); for (i=2; prime &&i<=k;i++) if (m%i==0) //m 被某除数整除 prime=0; //则不是质数 return prime; //用 return 语句返回 } //函数体结束</pre>	<pre>#include<stdio.h> #include<math.h> int main() { int m,i,k,prime=1; scanf("%d",&m); if (m==1) prime=0; k=(int)sqrt(m); for (i=2; prime &&i<=k;i++) if (m%i==0) prime=0; if (prime) printf("%d is a prime\n",m); else printf("%d is not a prime\n",m); return 0; }</pre>

这种方法还延伸到 C++入门知识的介绍中, 将 C 和 C++的输入输出对比(见表 3)。

表 3

(引用书中表 13.2) C++风格与 C 风格的字符的输入/输出

程序 13.2 字符输入/输出的 C++代码	程序 13.2 字符输入/输出的 C 代码
<pre>#include<iostream.h> int main() { char ch; cout<<"Input a character:"; cin>>ch; cout<<ch<<'<<(int)ch<<endl; return 0; }</pre>	<pre>#include<stdio.h> int main() { char ch; printf("Input a character:"); scanf("%c",&ch); printf("%c-%d\n",ch,ch); return 0; }</pre>
<p><运行结果> Input a character:A<回车> A-65</p>	<p><运行结果> Input a character:A<回车> A-65</p>

五、本书第 1 章独具特色。为了引入高级程序设计语言, 从最低级的机器语言开始, 重点讲解操作码、操作数、子程序调用过程、断点地址、入口地址等概念, 使学生了解程序最底层的调用和运行基于什么原理, 从而很自然地引出为什么需要发展到高级程序设计语言。用计算机技术

九、本书课后习题量少，但覆盖每章的主要知识点，参考了国家和江苏省等级考试的题目要求。在配套的辅导教材《C 语言程序设计习题解析与实验指导》中不仅配有主教材完整的习题解析，更提供了丰富的补充习题，帮助学生全面掌握 C 语言的知识。同时还精心设计了 10 个配套实验，帮助学生及时巩固和消化知识。

本书的学生用和教学用两种多媒体教学课件和书中所有的源代码可以从人民邮电出版社的教学服务与资源网 (<http://www.ptpedu.com.cn>) 免费下载。

本书是作者多年教学研究和经验的积累，更是课程组集体智慧的结晶。本书第 2 章由张霞和刘洁编写，第 3 章由刘志红和丛莹编写，第 4 章由郭剑和吴敏编写，第 8 章由许棣华和张勤编写，其余章节由朱立华和王立柱共同编写。朱立华负责全书的统稿工作。

由于作者水平有限，书中难免存在不当之处。读者如有问题或发现错误，欢迎直接与作者联系，作者将不胜感激。

作者 E-mail 地址为：zhulh@njupt.edu.cn、data_structure@sohu.com。

作者

2009 年 5 月

目 录

第 1 章 计算机的组成与程序设计

语言1

- 1.1 计算机的组成及工作过程.....1
- 1.2 机器语言程序设计.....5
- 1.3 机器语言的局限性.....6
- 1.4 高级程序设计语言与 C 语言.....7
- 1.5 C 语言程序及其开发.....9
- 本章小结.....10
- 习题11

第 2 章 基本数据类型.....12

- 2.1 常量与变量.....12
- 2.2 整型.....17
- 2.3 字符型.....22
- 2.4 实型.....26
- 2.5 typedef 类型名.....29
- 本章小结.....29
- 习题30

第 3 章 表达式与运算符.....33

- 3.1 表达式与算术运算符.....33
- 3.2 关系运算符.....34
- 3.3 逻辑运算符.....36
- 3.4 自增自减运算符.....38
- 3.5 位运算符.....39
- 3.6 条件运算符.....41
- 3.7 赋值和复合赋值运算符.....42
- 3.8 逗号运算符.....44
- 3.9 内部类型转换.....44
- 本章小结.....45
- 习题46

第 4 章 程序流程控制.....47

- 4.1 算法与语句.....47
 - 4.1.1 算法及其描述方法.....47

- 4.1.2 C 语言的语句分类.....49

- 4.2 顺序结构.....49
- 4.3 选择结构.....50
 - 4.3.1 if~else 语句.....50
 - 4.3.2 switch~case 语句.....54
- 4.4 循环结构.....56
 - 4.4.1 for 语句.....56
 - 4.4.2 while 语句.....59
 - 4.4.3 do~while 语句.....61
 - 4.4.4 循环嵌套.....64
- 4.5 其他流程控制语句.....67
 - 4.5.1 break 语句.....68
 - 4.5.2 continue 语句.....68
 - 4.5.3 goto 语句.....69
- 本章小结.....70
- 习题71

第 5 章 一级指针与一维数组.....75

- 5.1 一级指针类型.....75
 - 5.1.1 直接引用与间接引用.....75
 - 5.1.2 地址的类型——指针类型.....76
 - 5.1.3 指针的基本操作——间接引用.....77
 - 5.1.4 指针的基本操作——加减一个整数.....78
 - 5.1.5 指针的其他基本操作.....79
- 5.2 一维数组类型.....79
 - 5.2.1 一维数组的定义.....79
 - 5.2.2 一维数组的初始化.....80
 - 5.2.3 一维数组名的双重含义.....84
- 5.3 一级指针变量与一维数组.....85
 - 5.3.1 指针变量.....85
 - 5.3.2 指针变量定义中的“*”的位置.....87
 - 5.3.3 数组指针的基类型和指针变量的基类型相同.....87

5.3.4 数组指针的基类型和指针变量的 基类型不同	88	6.6.2 返回值与返回地址	128
5.3.5 一个变量等价于一个长度为 1 的 数组	89	6.6.3 返回值与传地址	130
5.3.6 取址运算和间接引用运算互为 逆运算	89	6.7 函数指针	131
5.3.7 “野”指针和 void 指针	90	6.8 递归	133
5.4 移动下标与移动指针的比较	91	本章小结	136
5.5 一维数组应用举例	93	习题	137
5.5.1 查找一个数据元素	93	第 7 章 模块化程序设计相关	141
5.5.2 插入一个数据元素	94	7.1 全局外部函数	141
5.5.3 删除一个数据元素	96	7.2 静态外部函数	142
5.5.4 选择法排序	97	7.3 全局外部变量	143
本章小结	98	7.4 静态外部变量	144
习题	99	7.5 const 常类型	145
第 6 章 函数	103	7.5.1 const 常量	145
6.1 函数定义与调用	103	7.5.2 指向 const 常量的指针	146
6.2 函数声明	107	7.5.3 const 常量指针	148
6.3 函数定义与调用举例	108	7.5.4 指向 const 常量的 const 常量 指针	150
6.3.1 判断质数	108	7.6 编译预处理	150
6.3.2 求最大公约数	110	7.6.1 无参宏指令	150
6.3.3 在数组中查找一个元素	111	7.6.2 带参宏指令	151
6.3.4 选择法排序	112	7.6.3 条件编译指令	152
6.3.5 冒泡法排序	113	7.6.4 文件包含指令	153
6.4 函数调用与变量的存储类别	115	7.7 模块化程序设计方法简介	155
6.4.1 自动局部变量	115	本章小结	156
6.4.2 静态局部变量	117	习题	157
6.4.3 外部变量	119	第 8 章 结构、联合、枚举	159
6.4.4 寄存器变量	120	8.1 结构	159
6.5 动态空间管理	120	8.1.1 结构定义	159
6.5.1 一维动态空间的申请与 释放	120	8.1.2 结构指针	162
6.5.2 “耗尽”动态空间资源的 实验	122	8.1.3 结构数组	162
6.5.3 一种“野”指针的实验	122	8.1.4 结构的嵌套	164
6.5.4 筛选法求质数	124	8.1.5 结构型返回值和地址调用	166
6.5.5 约瑟夫环问题	125	8.2 结构应用程序举例	166
6.6 关于函数调用的深入讨论	126	8.2.1 学生记录排序	166
6.6.1 传值与传地址	126	8.2.2 洗牌	168
		8.3 联合	170
		8.4 枚举	172
		本章小结	173

习题	174	11.5 文件的复制	224
第 9 章 字符串	177	本章小结	225
9.1 字符串的定义与存储	177	习题	226
9.2 字符串的赋值和输出	178	第 12 章 高级程序设计	228
9.2.1 字符串赋值	178	12.1 单链表	228
9.2.2 字符串的输入/输出处理	179	12.1.1 单链表结点结构的定义	229
9.3 字符串处理函数的原型	180	12.1.2 单链表的建立	229
9.4 字符串处理函数的实现	181	12.1.3 单链表的遍历	231
9.5 字符串应用程序举例	183	12.1.4 单链表的查找	231
9.5.1 数字串转换函数的应用	183	12.1.5 单链表的插入	232
9.5.2 判断回文	185	12.1.6 单链表的删除	233
本章小结	186	12.1.7 单链表的逆置	234
习题	187	12.1.8 单链表操作的完整程序	235
第 10 章 二维数组与指针	191	12.1.9 其他类型的单链表	244
10.1 二维数组的定义与赋值	191	12.2 学生成绩档案管理系统的设计与 实现	245
10.2 二维数组与一维数组	194	12.2.1 数据类型的定义	246
10.3 二维数组名的双重含义	197	12.2.2 为结点类型定制的基本 操作	247
10.4 二维数组与行指针	197	12.2.3 用二进制文件实现数据的 永久保存	248
10.5 二维数组与列指针	199	12.2.4 用两级菜单四层函数实现 系统	250
10.6 一维指针数组与二级指针	201	本章小结	260
10.7 一维指针数组与二维数组	203	习题	260
10.8 二维动态空间的申请与释放	205	第 13 章 C++初步	262
10.9 二维(数组)指针和一维(数组) 指针的相互转化	206	13.1 输入/输出	262
10.10 以二级指针为形参的 main 函数	207	13.2 结构	266
本章小结	208	13.3 内联函数	266
习题	209	13.4 默认函数	267
第 11 章 流与文件	212	13.5 函数重载	268
11.1 文件指针	212	13.6 运算符重载	268
11.2 文件的打开与关闭	213	13.7 引用	270
11.3 文件的读写	216	13.8 布尔型	272
11.3.1 字符的读写	216	本章小结	273
11.3.2 字符串的读写	218	习题	274
11.3.3 无格式读写	219		
11.3.4 格式读写	221		
11.4 文件的随机访问	223		

附录 A 常用字符与 ASCII 码对照表	275	附录 F C 语言程序设计常见错误及解决方案	286
附录 B C 语言的关键字	277	附录 G 命名规则	294
附录 C Visual C++ 下各数据类型所占字节数及取值范围	278	附录 H C 语言的发展简史	296
附录 D C 语言运算符的优先级与结合性	279	附录 I 函数 printf 的格式转换说明符	297
附录 E 常用的 ANSI C 标准库函数	281	附录 J 函数 printf 的格式修饰符	298
		参考文献	299

第 1 章

计算机的组成与程序设计语言

一个完整的计算机系统由硬件系统和软件系统组成。硬件系统是计算机实现自动控制与运算的物质基础，软件系统加载在硬件系统之上控制硬件完成各种功能。无论是系统软件还是应用软件，都是用程序设计语言编写的。程序设计语言的发展经历了从机器语言到汇编语言再到各种高级程序设计语言的过程。这个过程是有规律可循的，学习程序设计语言就是要发现和认识这个规律。机器语言是这个发展过程的始点，它包含着程序设计语言未来发展的一切细胞。而 C 语言是这个过程中的重要阶段，它不仅是计算机软件设计与开发的主流语言之一，也是认识和深入掌握其他程序设计语言的基础。

1.1 计算机的组成及工作过程

计算机处理存储的数据。可以说，存储和处理是一个整体：存储是为了处理，处理需要存储。“存储和处理的整体性”的最初表达是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，其主要内容为：

- (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，不妨用二进制代码表示；
- (3) 程序 and 数据的存储形式可以完全相同；

(4) 程序本身也可包含数据，即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的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。具体含义是：“将地址为 1000H 存储单元中的数据放到中央处理器中的寄存器 A 中”。

冯·诺依曼型计算机提出硬件由五部分组成，即运算器、控制器、存储器、输入设备和输出设备，其系统结构如图 1.1 所示。

运算器能够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。存储器存放程序指令及原始数据。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（Central Processing Unit, 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 1 个字节（1B）。1 个字节有 8 个二进制位，

存放 8 位二进制信息。每个字节有一个编号，称为地址（码）。

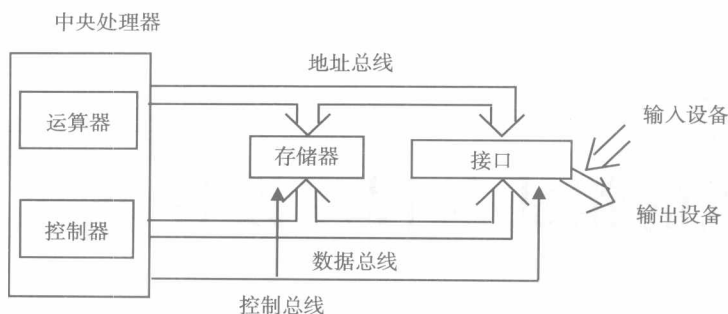


图 1.1 冯·诺依曼型计算机的系统结构框图

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，就应该有多少个地址。反之，有多少个地址，就有多少座房子。地址总线的多少决定了地址个数的多少，也就是字节的多少。设有 16 根地址总线，如果每根地址总线有脉冲信号表示 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} 个字节。

例如，1000000000000000 表示第 1 根地址总线有脉冲信号，其他地址总线没有脉冲信号，它代表一个二进制地址。这个地址可以用一个 4 位十六进制数简单的表示，即 8000H（H 代表十六进制）。4 位二进制数共有 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 位二进制数简洁地表示为一个 4 位十六进制数。例如 0011000000001111 对应 300fH。使用十六进制数表达仅仅为了简洁。类似地，如果有 32 根地址总线，那么 32 位二进制地址可以用 8 位十六进制数简洁地表示。

表 1.1 4 位二进制数和 1 位十六进制数的一一对应表

二进制数	十六进制数	二进制数	十六进制数	二进制数	十六进制数	二进制数	十六进制数
0000	0	0100	4	1000	8	1100	c (或 C)
0001	1	0101	5	1001	9	1101	d (或 D)
0010	2	0110	6	1010	a (或 A)	1110	e (或 E)
0011	3	0111	7	1011	b (或 B)	1111	f (或 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。CPU 通过地址总线发出相应的地址，选中存储器的该地址对应的即存储单元，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增加，多了就会影响速度。为了解决这个矛盾，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。

CPU 中的寄存器有两类，即通用寄存器和专用寄存器。B、C、D、E、H、L 是 8 位通用寄存器，它是运算器的组成部分，用来暂存操作数及运算的中间结果。它们可以组合使用作

为 16 位通用寄存器，即 BC、DE、HL。B'、C'、D'、E'、H'、L' 是它们的备用寄存器（或辅助寄存器）。



图 1.2 Z80 编程模型图

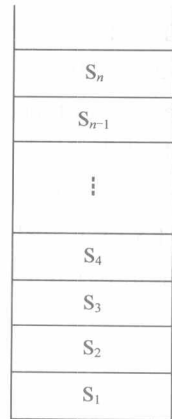


图 1.3 栈的结构示意图

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，只有与累加器 A 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。A' 是累加器 A 的备用寄存器。

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。F' 是标志寄存器 F 的备用寄存器。

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，否则 CPU 都是通过 PC 顺序地提取指令。例如，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，了解计算机的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 1.5(a) 所示。注意，y 不是数据，只是地址为 3002H 的存储单元的标识符，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。

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 单元中的数据放入累加器 A
03H	3001H	将地址为 3001H 单元中的数据与累加器 A 中的数据相加, 结果保留在寄存器 A
02H	3002H	将累加器 A 中的数据存入地址为 3002H 的单元
07H		停机

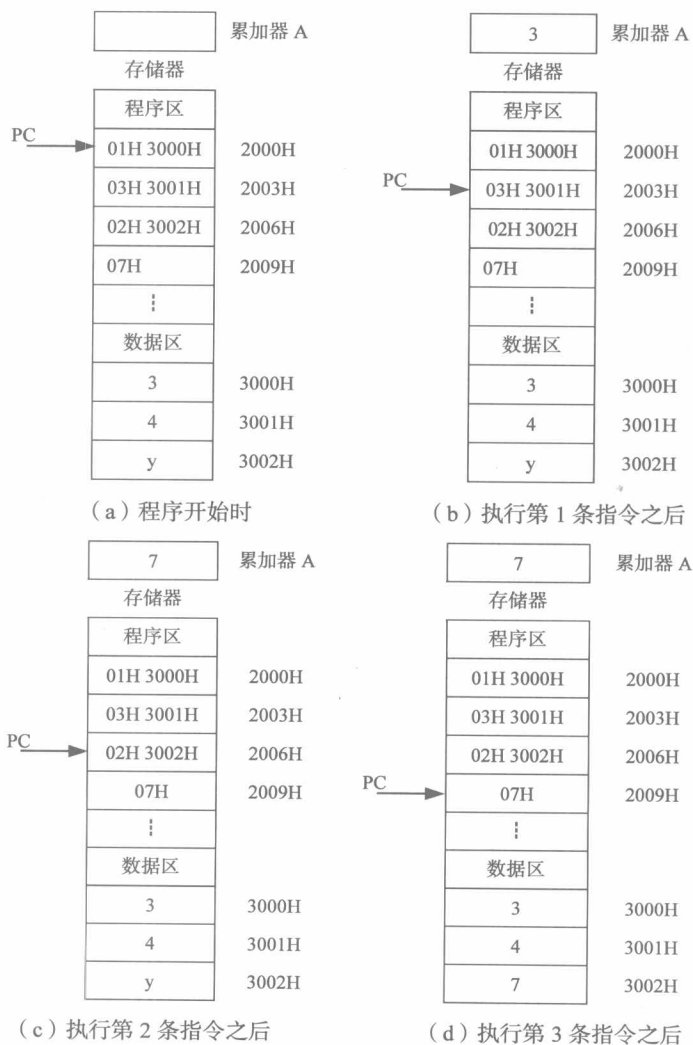
这组指令依次存储在存储器中的程序存储区, 地址分别为 2000H、2003H、2006H 和 2009H, 前 3 条指令各占 3 个字节, 第 4 条指令占一个字节。一条指令 (013000H) 的实际存储如图 1.4 所示。

01H	2000H
00H	2001H
30H	2002H

图 1.4 指令 013000H

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 1.5 演示了指令执行的结果, 第 4 条指令执行之后程序停止。

图 1.5 (a)~图 1.5 (d) 演示了指令执行的过程, 第 3 条指令执行结束后, 地址为 30002H 的存储单元中存放了运算结果 7, 第 4 条指令执行之后程序停止。

图 1.5 求和程序 $y=3+4$ 的执行过程示意图