

# 标准C语言 程序设计及应用

周纯杰 刘正林 何顶新 周凯波 编著

华中科技大学出版社  
<http://press.hust.edu.cn>

# 标准C语言 程序设计及应用

周纯杰 刘正林 何顶新 周凯波 编著

TP312C  
zh35

华中科技大学出版社  
<http://press.hust.edu.cn>

图书在版编目(CIP)数据

标准 C 语言程序设计及应用/周纯杰 刘正林 何顶新 周凯波 编著  
武汉:华中科技大学出版社,2005年3月

ISBN 7-5609-3343-2

- I. 标…
- II. ①周… ②刘… ③何… ④周…
- III. C 语言-程序设计
- IV. TP312

标准 C 语言程序设计及应用

周纯杰 刘正林 编著  
何顶新 周凯波

责任编辑:沈旭日

封面设计:潘群

责任校对:陈骏

责任监印:熊庆玉

出版发行:华中科技大学出版社

武昌喻家山 邮编:430074 电话:(027)87557437

录排:华中科技大学惠友文印中心

印刷:湖北省新华印务有限公司

开本:787×1092 1/16

印张:22.75

字数:480 000

版次:2005年3月第1版

印次:2005年3月第1次印刷

定价:29.80元

ISBN 7-5609-3343-2/TP·557

(本书若有印装质量问题,请向出版社发行部调换)

## 内 容 简 介

本书以 ANSI C++ 标准为准则，采取循序渐进、突出重点、分散难点的编写方法，系统地讲授作为 ANSI C++ 内核的 C 语言的基本语法和程序设计方法。

全书共分 11 章：C 语言基础知识，数据类型、运算符和表达式，程序和流程控制，变量的存储类型，数组，指针，函数，结构和联合，文件，编译预处理，C 语言的实际应用。每章都有例题、小结、习题，便于读者学习与复习。这些例题都是精选的，并经过上机检验的。为了满足读者上机练习的需要，书后还给出了 5 个附录，介绍 C 语言中常用库函数、Borland C++ V 3. 1 集成环境下运行 C 语言程序的方法，以及在上机过程中常见的编译错误及其原因。

本书内容新颖、通俗易懂，非常重视对学生编程思想和编程规范的培养，是学习 C 语言的理想教材。



# 前 言

---

C语言是近十几年来在国内外得到迅速推广应用的一种计算机语言。C语言具有功能丰富、表达力强、使用灵活方便、应用面广、目标效率高以及可移植性好等特点；既具有高级语言的优点，又具有低级语言的许多特点。因此，C语言在系统软件和应用软件的开发中得到了广泛的应用。

现在，许多高等学校，不仅在计算机专业开设了C语言课程，而且在非计算机专业也开设了C语言课程，并且作为第一门计算机语言课程开设，因此，在C语言的教学过程中应注意教学内容的循序渐进，另外要特别注意对学生的编程思想和编程规范的培养。2000年，在刘正林教授的主持下，曾经出版《最新C语言程序设计教程》。该书出版以后，受到了广大读者的好评，如视角新颖、概念清楚、内容丰富等，在国内被多所高校相继在教学中采用。为了适应当前高校计算机教育的发展和教学改革的需要，我们综合多年教学经验，在刘正林教授的支持和帮助下，编写了这本《标准C语言程序设计及应用》教材。该教材的特点体现在如下几方面。

(1) 以最新ANSI C++标准为准则，扬弃了一些C语言老版本的一些非标准内容。例如，函数原型的说明、void指针等。

(2) 充分考虑到初学者的特点，整个教材采取了循序渐进、逐层推进的编写方式，如先介绍数据和表达式，再介绍简单程序设计及流程控制；为了分散难点，把存储类型单独作为一章介绍，介绍了复杂数据类型数组和指针之后，再介绍函数等。

(3) 按照程序设计的特点，将指针放在函数前面讲述。在C语言程序设计中，指针的地位非常重要，它是为了提高程序的设计效率和程序的模块化设计而引入的，指针的一个主要的功能为函数参数传递服务，所以先介绍指针，再介绍函数，符合程序的设计特点。这已我们从多年的教学实践中得到了证明。

(4) 非常重视学生的编程思想和编程规范的培养。在整个教材中，不论是一个简单的程序（一个函数，几条语句），还是相对复杂的程序，都充分体现编程思想并力求做到编程（书写和编程设计）规范。

(5) 考虑到第2章的内容涉及输入/输出函数，所以在第1章中先介绍了输入/输出函数的简单应用，在学生对C语言有了一定认识之后，再对输入/输出函数的复杂应用进行了进一步的讨论。对于学习过程中一些基本概念容易出错的地方，另辟章节介绍，例如，第2章最后一节和第6章的最后一节将数据与指针常见问题进行了分析，这对于学生掌握有关概念是有好处的。

(6) 我们认为，学习计算机语言的最终目的是能够亲自动手编程，所以在教材中非常注意引导学生如何进行程序设计，包括简单的程序设计和复杂的程序设计。另外，在教材中我们还精选了大量例题，这些例题实用性强，都经过了上机验证。最后一章，还针对一些实际

应用进行介绍。

(7) 为满足初学者上机练习的需要,在附录Ⅲ中分类介绍了 C 语言中的常用库函数,在附录Ⅳ中介绍了 Borland C++ V3.1 集成环境下运行 C 语言程序的方法,在附录Ⅴ介绍了上机过程中常见的编译错误及可能的出错原因及解决办法,这些对于初学者上机编程都是非常有帮助的。

(8) 考虑到计算机等级考试等的需要,我们按照相关考试的要求,组织了大量的习题,这些习题覆盖了各种考试的题型,所以本书对于计算机等级考试也是一本极好的参考书。

在撰写本书的过程中,既参考了《The C programming Language (Second Edition)》(Brain W 编著)和《最新 C 语言程序设计》(刘正林等编著)、《C 语言程序设计教程》(曹化工等编著)及《C 语言程序设计》(谭浩强编著)等国内外多种教材,也融入了作者在华中科技大学电子与信息学院多年从事教学和科研的实践经验和体会,同时还吸收了同行专家学者的意见和建议。

本书由周纯杰副教授和刘正林教授进行整体规划和完成统稿工作,其中第 1 章和附录Ⅳ由刘正林教授撰写,第 2 章、第 9 章和第 10 章由周凯波博士撰写,第 3 章、第 4 章、第 5 章及第 11.3 节由何顶新副教授撰写,第 6 章、第 7 章、第 8 章、第 11.1 与 11.2 节和本书的其余部分由周纯杰副教授撰写。

本书的出版得到了华中科技大学信息学院、控制系以及教务处的领导和同事们的关心和支持,华中科技大学出版社的有关同志为其出版也付出了辛勤的劳动,硕士研究生张涛对资料的整理和程序的验证进行了有效的工作,另外硕士研究生周孝慧、刘颢、宋明权、陶志东、熊锐也进行了部分资料整理和程序验证工作。在此一并表示感谢!

由于作者水平有限,书中疏漏或错误之处恳请广大读者批评指正。

编 者

2004 年 12 月于华中科技大学

# 目 录

---

<b>第 1 章 概论</b> .....	(1)
1.1 C 语言的入门知识 .....	(1)
1.1.1 二进制编码系统 .....	(1)
1.1.2 微型计算机硬件的基本组成 .....	(2)
1.1.3 计算机系统的层次结构 .....	(5)
1.2 C 语言的发展及特点 .....	(9)
1.2.1 C 语言的发展过程 .....	(9)
1.2.2 C 语言的特点 .....	(10)
1.3 C 语言程序的书写格式和结构特点 .....	(13)
1.3.1 C 语言程序的书写格式 .....	(13)
1.3.2 C 语言程序的结构特点 .....	(15)
1.4 C 语言的基本语法单位 .....	(17)
1.4.1 标识符 .....	(17)
1.4.2 关键字 .....	(18)
1.4.3 分隔符 .....	(19)
1.4.4 常量 .....	(19)
1.5 简单的输入/输出 .....	(19)
1.5.1 格式化输入/输出函数 .....	(20)
1.5.2 字符输入/输出函数 .....	(22)
1.6 运行 C 程序的一般步骤 .....	(23)
小结 .....	(24)
习题一 .....	(24)
<b>第 2 章 数据类型、运算符和表达式</b> .....	(25)
2.1 数据类型 .....	(25)
2.2 常量和变量 .....	(25)
2.2.1 常量 .....	(26)
2.2.2 变量 .....	(30)
2.3 运算符和表达式 .....	(33)
2.3.1 表达式 .....	(33)
2.3.2 算术运算符与算术表达式 .....	(33)

2.3.3	关系运算符与关系表达式	(37)
2.3.4	逻辑运算符与逻辑表达式	(38)
2.3.5	自增和自减运算	(39)
2.3.6	赋值运算符与赋值表达式	(40)
2.3.7	条件运算符与条件表达式	(41)
2.3.8	逗号运算符与逗号表达式	(42)
2.4	位运算	(42)
2.4.1	按位与运算符“&”	(42)
2.4.2	按位或运算符“ ”	(43)
2.4.3	按位异或运算符“^”	(44)
2.4.4	二进制左移运算符“<<”	(45)
2.4.5	二进制右移运算符“>>”	(45)
2.4.6	按位取反运算符“~”	(46)
2.5	运算符的优先级	(46)
2.6	格式化输入/输出函数的进一步讨论	(47)
2.6.1	格式化输出函数 printf	(47)
2.6.2	scanf 函数(格式输入函数)	(50)
2.7	常见问题分析	(51)
	小结	(54)
	习题二	(54)
<b>第 3 章</b>	<b>程序和流程控制</b>	<b>(58)</b>
3.1	C 语言程序的版式和语句	(58)
3.1.1	C 语言程序的版式	(58)
3.1.2	C 语言的语句	(60)
3.2	结构化程序设计和流程控制	(61)
3.2.1	结构化程序设计	(61)
3.2.2	C 语言的流程控制语句和辅助控制语句	(62)
3.3	if 语句	(63)
3.3.1	if 语句的标准形式	(63)
3.3.2	条件分支嵌套	(64)
3.4	switch 多分支选择语句	(67)
3.5	循环控制	(71)
3.5.1	while 语句	(71)
3.5.2	for 语句	(72)
3.5.3	do-while 语句	(74)
3.5.4	从一重循环到多重循环	(75)



3.6 辅助控制语句 .....	(78)
3.6.1 break 语句 .....	(78)
3.6.2 continue 语句 .....	(80)
3.6.3 goto 语句和标号 .....	(81)
3.7 典型程序编写方法举例 .....	(82)
3.7.1 典型问题一 .....	(83)
3.7.2 典型问题二 .....	(86)
3.7.3 典型问题三 .....	(92)
小结 .....	(94)
习题三 .....	(95)
<b>第4章 变量的存储类型</b> .....	<b>(98)</b>
4.1 概述 .....	(98)
4.2 自动型变量 auto .....	(99)
4.3 寄存器型变量 register .....	(103)
4.4 外部参照型变量 extern .....	(105)
4.5 静态型变量 static .....	(108)
小结 .....	(111)
习题四 .....	(112)
<b>第5章 数组</b> .....	<b>(114)</b>
5.1 一维数组的定义和应用 .....	(114)
5.2 二维数组 .....	(122)
5.3 字符数组 .....	(127)
5.4 程序设计举例 .....	(132)
小结 .....	(136)
习题五 .....	(136)
<b>第6章 指针</b> .....	<b>(141)</b>
6.1 指针概念 .....	(141)
6.1.1 变量的地址 .....	(141)
6.1.2 指针变量 .....	(142)
6.2 指针变量的定义和使用 .....	(143)
6.2.1 指针变量的定义及初始化 .....	(143)
6.2.2 指针的使用 .....	(146)
6.3 指针运算 .....	(148)
6.3.1 指针的算术运算 .....	(148)

6.3.2	关系运算	(152)
6.3.3	指针的赋值运算	(152)
6.4	指针与数组及字符串	(155)
6.4.1	指针与数组	(155)
6.4.2	字符指针与字符串	(158)
6.5	指针数组和多级指针	(159)
6.5.1	指针数组	(159)
6.5.2	多级指针	(163)
	小结	(166)
	习题六	(167)

## 第 7 章 函数 (171)

7.1	结构化程序设计与 C 语言程序结构	(171)
7.1.1	结构化软件及其优越性	(171)
7.1.2	C 语言程序的结构	(173)
7.2	函数的定义和调用	(173)
7.2.1	函数的定义	(174)
7.2.2	函数的调用	(176)
7.2.3	参数数目可变的函数	(177)
7.3	函数间的数据传递	(177)
7.3.1	使用函数参数在函数间传递数据	(178)
7.3.2	使用返回值传递数据	(181)
7.3.3	使用全局变量传递数据	(182)
7.4	数组与函数	(183)
7.4.1	数组元素作为函数实参	(184)
7.4.2	一维数组名作为函数参数	(185)
7.4.3	多维数组名作为函数参数	(188)
7.5	字符串与函数	(192)
7.5.1	常见字符串处理库函数及其使用	(192)
7.5.2	单个字符串的处理	(195)
7.5.3	多个字符串的处理	(198)
7.6	指针型函数	(201)
7.7	递归函数	(206)
7.8	指向函数的指针	(208)
7.9	带参数的函数 main	(213)
7.10	程序设计综合举例	(215)
	小结	(227)

习题七 .....	(227)
<b>第 8 章 结构和联合 .....</b>	<b>(232)</b>
8.1 结构的定义以及结构变量的定义和使用 .....	(232)
8.1.1 结构的定义 .....	(232)
8.1.2 结构变量的定义 .....	(233)
8.1.3 结构变量的使用形式和初始化 .....	(234)
8.2 结构数组与结构指针 .....	(237)
8.2.1 结构数组 .....	(237)
8.2.2 结构指针 .....	(240)
8.3 结构在函数间的传递 .....	(244)
8.4 位字段结构 .....	(249)
8.5 联合 .....	(251)
8.6 类型定义语句 typedef .....	(254)
8.6.1 用 typedef 语句定义新类型名 .....	(254)
8.6.2 新类型名的应用 .....	(255)
8.7 枚举类型 .....	(257)
8.7.1 枚举类型的定义和枚举变量的说明 .....	(257)
8.7.2 枚举类型的应用 .....	(259)
8.8 综合举例 .....	(260)
小结 .....	(263)
习题八 .....	(263)
<b>第 9 章 文件 .....</b>	<b>(267)</b>
9.1 文件的基本概念 .....	(267)
9.1.1 文本文件与二进制文件 .....	(267)
9.1.2 缓冲型文件系统 .....	(267)
9.2 文件类型指针 .....	(268)
9.3 文件的打开与关闭 .....	(269)
9.4 常用文件读/写函数 .....	(271)
9.5 文件的随机读/写 .....	(279)
9.6 文件检测函数 .....	(281)
小结 .....	(282)
习题九 .....	(282)
<b>第 10 章 编译预处理 .....</b>	<b>(283)</b>
10.1 宏定义 .....	(283)

10.2 文件包含 .....	(287)
10.3 条件编译 .....	(288)
小结 .....	(290)
习题十 .....	(290)
<b>第 11 章 C 语言的实际应用 .....</b>	<b>(293)</b>
11.1 图形程序设计 .....	(293)
11.1.1 控制图形系统的主要函数 .....	(293)
11.1.2 基本作图函数 .....	(295)
11.1.3 图形方式下的文本常见操作函数 .....	(299)
11.1.4 综合应用举例 .....	(302)
11.2 中断程序设计 .....	(303)
11.2.1 中断技术 .....	(303)
11.2.2 用 C 语言编写中断服务程序的方法 .....	(304)
11.2.3 中断服务程序综合应用举例 .....	(306)
11.3 链表的 C 语言编程 .....	(310)
11.3.1 单链表的构造 .....	(311)
11.3.2 单链表的操作 .....	(314)
小结 .....	(315)
习题十一 .....	(315)
<b>附录 I ASCII 码表 .....</b>	<b>(317)</b>
<b>附录 II C 语言中的关键字 .....</b>	<b>(318)</b>
<b>附录 III C 语言常用的库函数 .....</b>	<b>(318)</b>
<b>附录 IV Borland C++V3.1 的使用 .....</b>	<b>(323)</b>
<b>附录 V Borland C++V3.1 常见编译错误信息 .....</b>	<b>(349)</b>
<b>主要参考文献 .....</b>	<b>(352)</b>

# 第 1 章

## 概 论

C 语言和由它发展而来的 C++ 是当今流行的两种程序设计语言,是两种重要的编程工具。C 语言在最初设计时,是作为一种面向系统软件的开发语言,是用来代替汇编语言的,但是,由于它强大的生命力,后来在事务处理、科学计算、工业控制和数据库技术等各个方面都得到了广泛应用。即便进入到以计算机网络为核心的信息时代,C 语言仍然作为通用的汇编语言使用,用以开发软(件)、硬(件)结合的程序,如实时监控程序、系统控制程序和设备驱动程序等。当前,作为理工科各类专业本科生的计算机基础课程“C 语言程序设计”应按照最新的 ANSI(America National Standard Institute,美国国家标准局)C++ 标准的基础内核为准则来讲授,因为该标准已成为当今世界公认的 C++ 工业标准。本书将重点介绍 C 程序设计语言的主要语法和模块结构化基础知识,引导学生按新的 ANSI C++ 标准学习和编写 C 语言程序。为此,首先介绍一些计算机的基础知识,为讲解 C 和 C++ 做好必要的准备。

## 1.1 C 语言的入门知识

### 1.1.1 二进制编码系统

计算机只能识别和处理二进制码,它不仅要用二进制码表示数值,还要用二进制码表示其他各种信息,例如,字符、大小写英文字母、运算符和标点符号等;而各种字符只能用数个 bits 按一定规则进行不同的排列和组合所构成的二进制码来表示,这就是二进制编码。现在,英文字母、运算符和标点符号等字符都采用 ASCII 码(ASCII 是 American Standard Code for Information Interchange(美国国家标准信息交换码)的缩写),详见附录 I。该代码由 7 位(bit,简称 b)组成,可表示 128 个字符。由于微型计算机内存的一个存储单元是 8 个 bits,称为一个字节(byte,简称 B),因此,一个字节只能表示一个 ASCII 码字符,最高位设置为 0。图 1.1 列举了几个重要的 ASCII 码,其中, $b_5$  和  $b_4$  为 1 的字符(十六进制高位为 3)是数字字符, $b_6$  为 1 的字符(十六进制高位为 4)是英文大写字母, $b_5$  和  $b_6$  为 1 的字符(十六进制高位为 6)是英文小写字母。

对于多媒体计算机,把文本、图像和声音等信息按某种标准格式进行二进制编码,就可解决文本、图像和声音等的控制管理、存储、变换和传送等。行使这些功能的系统都统称为二进制编码系统。

	b <sub>7</sub>	b <sub>6</sub>	b <sub>5</sub>	b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	b <sub>0</sub>	
字符 NULL:	0	0	0	0	0	0	0	0	= 00(H)
数字字符 0:	0	0	1	1	0	0	0	0	= 30(H)
数字字符 9:	0	0	1	1	1	0	0	1	= 39(H)
英大字母 A:	0	1	0	0	0	0	0	1	= 41(H)
英大字母 B:	0	1	0	0	0	0	1	0	= 42(H)
英小字母 a:	0	1	1	0	0	0	0	1	= 61(H)
英小字母 b:	0	1	1	0	0	0	1	0	= 62(H)

图 1.1 '0'、'9'、'A'和'a'等字符的 ASCII 码

### 1.1.2 微型计算机硬件的基本组成

计算机发展到今天,虽然引起了多次世界范围内的技术革命,改变着人类生活的方方面面,但从它的基本运行原理看,与 1946 年问世的第一台电子计算机在结构上大同小异,都可统称为冯·诺依曼(Von Neumann)型计算机,仍然由机器的硬件顺序地执行一条一条指令来完成所规定的任务。机器的硬件由运算器、控制器、存储器、输入设备和输出设备等 5 个部分组成。运算器和控制器是计算机的核心,前者能完成对数据的各种运算,后者是整个计算机的指挥中心。微型计算机(MicroComputer,简称 MC 或  $\mu C$ )是按照电子计算机的基本工作原理,由借助微电子技术制造出来的大规模和超大规模集成电路发展起来的,它将运算器和控制器集成在一块大规模或超大规模集成电路芯片上,这种芯片称为中央处理单元,即 CPU(Central Processing Unit),再加上一个存储器和输入/输出(I/O)接口电路就构成一台“主机(Host)”,其基本结构如图 1.2 所示。

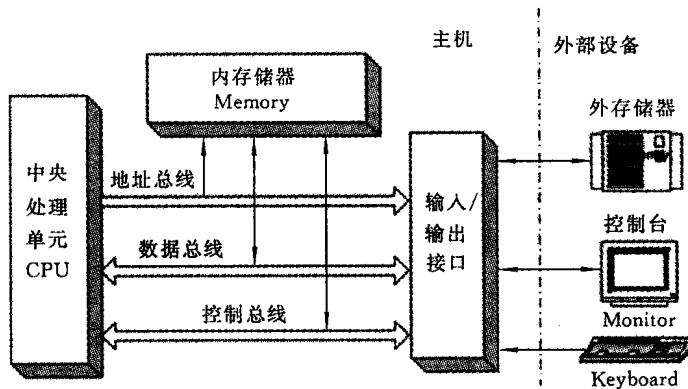


图 1.2 主机系统的基本结构示意图

#### 1. 中央处理单元 CPU

它是进行数据运算和信息处理的核心部件,其输出以 3 种总线,即地址总线( $A_0 \sim A_{31}$ )、数据总线(8 位机  $D_0 \sim D_7$ 、16 位机  $D_0 \sim D_{15}$ 、32 位机  $D_0 \sim D_{31}$ 、64 位机  $D_0 \sim D_{63}$ )和控制总线,与内存存储器(Memory)和输入/输出接口电路相连。内存存储器和输入/输出接口电路也利



用微电子技术集成在大规模和超大规模集成芯片上,并使用一些标准总线实现互连,如个人专用计算机,即 PC(Personal Computer)机以前使用的 PC 总线与 ISA(Industry Standard Architecture,工业标准体系结构)总线相同。1991 年美国著名的芯片制造厂商 Intel 公司提出的 PCI(Peripheral Component Interconnect,外设部件互连)总线已成为当前广泛流行的标准总线。若把 CPU、数量有限的内存储器 and 输入/输出接口电路集成在一块芯片上就构成了单片微型计算机,简称单片机,它适用于功能较简单、程序容量较小的微小型计算机系统。顺便指出,通常把 CPU 一次处理的二进制信息的位数(bits 数)称为计算机的字长,它与数据总线的宽度(根数)相等。

## 2. 存储器

计算机与人脑的记忆神经一样,有一个具有记忆功能的组成部分叫做存储器,它不仅能记忆题目和数据,而且还能记忆运算法则、计算步骤和“口诀”等。由于计算机只能识别二进制码信息,因此,还要用某种格式的二进制码设计一个控制计算机执行各种操作的指令系统,每条指令命令计算机执行某种操作,这些指令称为机器码指令(Machine Code)。存储器分为外存储器和内存储器,前者存储容量较大,通常放在主机外面做成计算机的外部设备,如图 1.2 所示的外存储器;而图 1.2 所示的内存储器则属于主机不可缺少的组成部分,它是主机内存放各种数据的一个大“仓库”,通常由具有“记忆”功能的电子部件组成。为了便于管理,把这个大仓库分成一个个“小房间”,称为存储单元,每个存储单元为一个字节(8b),给每个存储单元编上一个号码,称为地址码,实际上也是二进制码,这就像给大仓库的“小房间”编排一个门牌号码。正如上述的地址总线( $A_0 \sim A_{31}$ )的地址码为 32 位,可表示  $2^{32}$  个号码,它能够给出 40 亿个地址码,即

$$2^{32} = 2^2 \times 2^{30} \approx 4 \times 10^9 = 4G \quad (1G = 1000M = 10 \text{ 亿})$$

通常,用十六进制数来表示地址码要方便得多,如图 1.3 所示,若地址总线为  $A_0 \sim A_{15}$ ,即 16 位地址码, $A_0 \sim A_{15}$  都是二进制码,即取 0 和 1 两种状态值,则可用 4 位十六进制码表示。最高位表示  $A_{15} \sim A_{12}$ ,次高位表示  $A_{11} \sim A_8$ ,第 3 位表示  $A_7 \sim A_4$ ,最低位表示  $A_3 \sim A_0$ 。因此,按二进制和十六进制的对换关系,可将地址总线的  $A_0 \sim A_{15}$  表示成 4 位十六进制数 8001H~8004H(H 是 Hexadecimal 的第 1 个字母)。

	$A_{15}$	$A_{14}$	$A_{13}$	$A_{12}$	$A_{11}$	$A_{10}$	$A_9$	$A_8$	$A_7$	$A_6$	$A_5$	$A_4$	$A_3$	$A_2$	$A_1$	$A_0$			
8001H =	1	0	0	0	,	0	0	0	0	,	0	0	0	0	,	0	0	0	1
8002H =	1	0	0	0	,	0	0	0	0	,	0	0	0	0	,	0	0	1	0
8003H =	1	0	0	0	,	0	0	0	0	,	0	0	0	0	,	0	0	1	1
8004H =	1	0	0	0	,	0	0	0	0	,	0	0	0	0	,	0	1	0	0

图 1.3 用 4 位十六进制数表示地址总线的  $A_0 \sim A_{15}$

控制计算机进行各种操作和运算的命令称为指令,也是以二进制码形式来表示的,称为机器码指令。对于不同的 CPU,其机器码的编码规律是不同的,它们都有一套自己的机器码指令集,叫做指令系统。例如,Intel 公司生产的 MCS51 系列单片(微型计算)机的指令系统中,有如下指令:

机器码指令	汇编指令	操作内容
0000 ,0100(04H)	INC A	A + 1→A
0000 ,1110(0EH)	INC R1	R1+ 1→R1
.....		

因为计算机只懂得机器码指令,也就是二进制码,但机器码指令不便于阅读、理解和记忆,而且容易出错,所以促使人们使用助记符号来代替机器码指令,即对于每一条机器码指令,使用与其操作内容相联系的英文缩写字符串来表示。例如,INC A(INC 为单词 Increment(增量,加 1)的缩写),是把累加器 A 的内容(所存放的数据)增 1 后再放回 A。这种符号语言称为汇编语言,用它来编写程序比机器码指令要方便得多,但这类程序必须经过机器翻译(称为汇编)成用机器码指令表示的程序(称为目标程序或可重定位文件,以“.obj”作为文件的扩展名),计算机才能理解和执行。这种翻译工作是由一种专门的、称之为“汇编系统”的程序来完成的,然后通过一个“链接、装配程序”生成可执行文件,即以“.exe”为扩展名的文件。用高级语言如 C/C++ 编写的程序,也必须使用这些语言系统的编译器翻译(称为编译)成用机器码指令表示的程序即目标程序之后再链接、装配成可执行文件。计算机若要运行一个可执行文件,则要通过操作系统把该文件装配到内存存储器中,然后再从它的第 1 条机器码指令开始顺序往下执行,直到最后一条机器码指令执行完为止。

### 3. 输入/输出接口电路和外部设备

一个微型计算机系统的构成如图 1.4 所示,硬件部分除了主机以外还必须连接外部设备,因为它是人与机器进行交互(即沟通)的接口。其中,必备的输入设备是键盘,必备的输出设备为显示器,它们构成机器的“控制台(Console)”。CPU 通过输入/输出接口与外部设备相连,这些接口电路也是用地址总线、数据总线和控制总线与 CPU 连接起来的,并且也都利用微电子技术集成为与 CPU 配套的接口芯片。

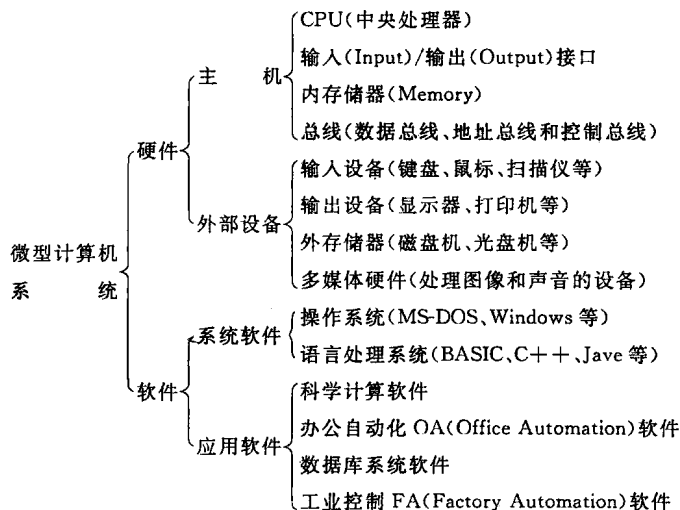


图 1.4 微型计算机系统的构成

微型计算机的软件分为系统软件和应用软件两大类,前者包含操作系统和语言处理系

统等,后者是计算机应用领域的大舞台,将在下一节详细讨论。

### 1.1.3 计算机系统的层次结构

正如图 1.5 所示,任何计算机系统必须包含硬件和在其上运行的软件。计算机硬件俗称为裸机(物理机器),它必须有软件的支持才能有效地运转并为人们所利用。所以计算机软件资源是计算机系统的重要组成部分。

若要将计算机应用于各个不同的专业领域,则必须对其硬、软件进行“二次开发”才能满足各应用领域的不同需求。为了有效地利用已有的计算机软件资源为二次开发服务,常将这些软件资源按图 1.5 所示的层次结构,构造成“用户开发平台”。随着计算机应用的广泛普及和计算机网络时代的到来,用户开发平台的硬、软件资源均成为对用户完全开放的国际工业标准,演变成当今世界大力提倡和广泛采用的开放系统开发平台(Open System Platform)。



图 1.5 计算机系统的层次结构图

不管是计算机技术的开发人员,还是计算机使用人员,了解开放系统开发平台的组成是很有必要的,下面从系统层次观点的角度,对照图 1.5 所示结构简要地加以介绍。

#### 1. 操作系统(OS, Operating System)

操作系统它是计算机软件的核心,是硬件的第一级扩充。在它的控制下,计算机的全部资源,如 CPU、内存储器、外部设备(显示器、磁盘机、光盘机、电子盘、打印机、扫描仪、X-Y记录仪等)和各种软件资源可以协调一致地工作。它可以有条不紊、高效率地管理和调度计算机的硬件设备和各种软件资源,使它们最大限度地发挥作用。操作系统还要协调控制许多可能“并发”执行的程序段,按照预先确定的控制策略合理地组织系统的工作流程,提高系统的执行效率,保护系统和用户的信息安全。显然,只有当计算机硬件配上了操作系统,计算机系统的整个工作过程才能实现全盘自动化,即具有自动调度、自动协调、自动管理和自动运行的能力,从而最大限度地减少用户手工操作的负担,并在工作过程中随时接受和处理用户提交的任务。目前,在微型计算机应用领域广为流行的有如下操作系统。

##### (1) MS-DOS

MS-DOS 是美国 Microsoft 公司为 IBM-PC 微型计算机开发的磁盘操作系统,后被美国 IBM 公司所选用,也称 PC-DOS,是单用户、单任务的操作系统。早已广泛地应用在个人专用微型计算机,即 PC 机上。Microsoft 公司自推出 V6.22 版本后,即宣称它是最终版本,相当于宣判了 DOS 的死刑,但要在应用领域让它销声匿迹还需要一段时间,且在 Windows 平台上还设置有 MS-DOS 方式,在需要运行老的 DOS 源程序时能与 DOS 版本兼容。

##### (2) Windows

Windows 是美国 Microsoft 公司开发的基于可视化图形的多任务、多窗口操作系统。