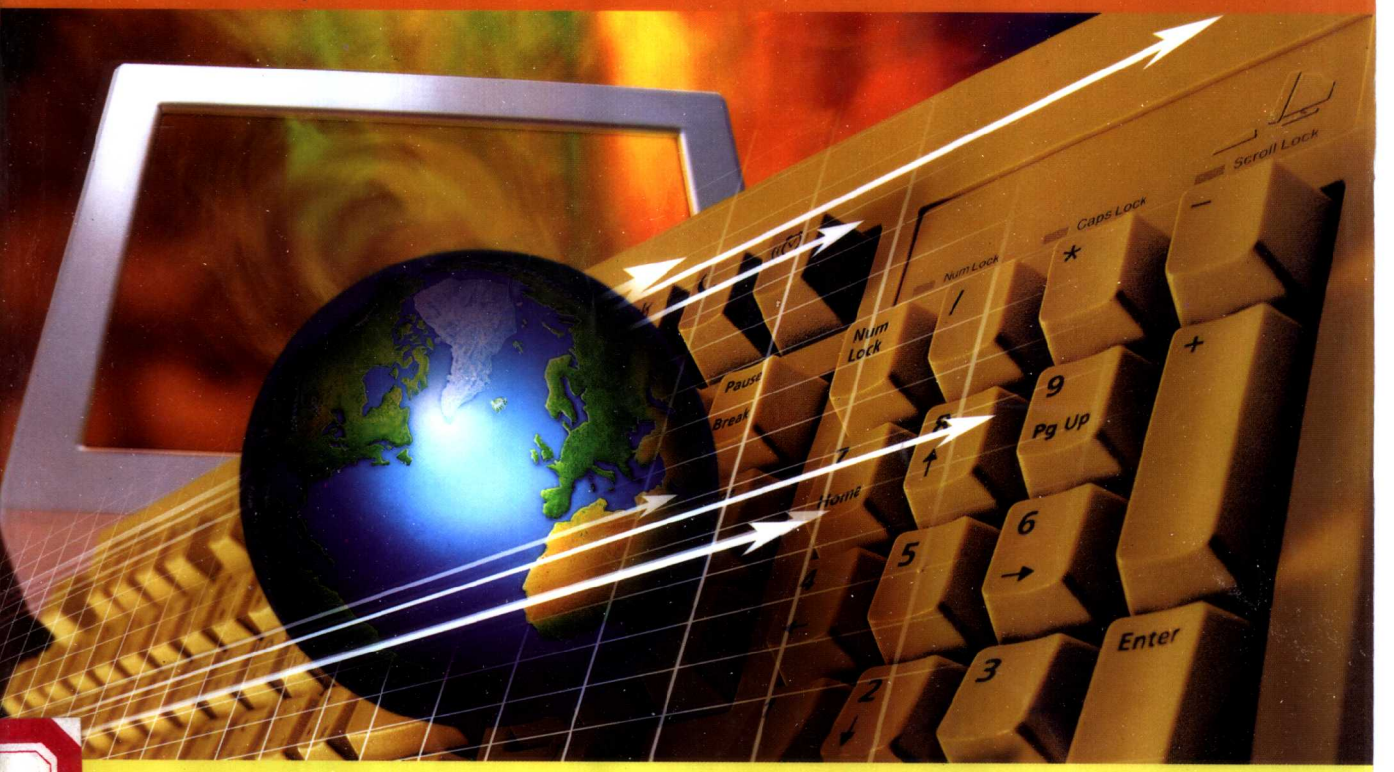
C语言程序设计教程

马德骏　张建宏　汤练兵　主编

C: YUYAN CHENGXU SHEJI JIAOCHENG



科学出版社
www.sciencep.com

C语言程序设计教程

[redacted]

C语言程序设计教程

马德骏　张建宏　汤练兵　主编

科学出版社
北京

内 容 简 介

本书面向非计算机专业初学程序设计的读者。全书共分十一章，前十章讲述了计算机的基础知识和C语言的基本知识、基本算法及基本的程序设计方法。第十一章介绍了一些实用的示例。

本书力求通俗易懂，便于自学。书中配有一定数量的习题，与之配套的《C语言程序设计实验与习题》一书中对其中大多数的习题给出了参考答案。

本书可作为大学本、专科非计算机专业学生学习C语言程序设计的教材，也可供计算机等级考试者和其他各类学习者使用和参考。

图书在版编目(CIP)数据

C语言程序设计教程/马德骏，张建宏，汤练兵主编. —北京：科学出版社，2002.12

ISBN 7-03-011100-1

Ⅰ.C… Ⅱ.①马…②张…③汤 Ⅲ.C语言-程序设计-教材

Ⅳ.TP312

中国版本图书馆CIP数据核字（2002）第107642号

责任编辑：冯青辰/责任校对：王望荣

责任印制：高 嵘/封面设计：李冬华 李 静

科学出版社出版

北京东黄城根北街16号

邮政编码：100717

http://www.sciencep.com

武汉大学出版社印刷总厂印刷

科学出版社出版 各地新华书店经销

*

2003年1月第 一 版 开本：787×1092 1/16

2003年1月第一次印刷 印张：13 3/4

印数：1—10 000 字数：339 000

定价：18.00元

（如有印装质量问题，我社负责调换）

前　言

计算机的文化、技术和应用水平是当今衡量一个人的知识水平和能力水平的重要标准之一。程序设计课程是计算机基本技能教育和能力培养的一个重要部分，它能培养学生利用计算机解决实际问题的思维方式和能力，也能为学生在后继课程以及工作中应用计算机解决实际问题打下基础。C 语言已成为大多数学校理工科专业所选的程序设计课程的教学语言，也是在计算机各种证书考试中为大多数人所选择的程序设计语言。

《C 语言程序设计教程》是根据我们长期教学实践经验为初学程序设计者编写的一本教材，也是我们教学的一个阶段性小结。本书共有十一章，前十章讲述了 C 语言的基本知识、基本算法和基本的程序设计方法，分别为：C 语言程序设计基础知识，基本数据类型及其运算，顺序结构程序设计，选择结构程序设计，循环结构程序设计，数组，函数，指针，结构体、共用体和枚举，文件。第十一章介绍了一些实用的示例，旨在增强学习者的综合应用能力，使读者在 C 语言的学习过程中能够再上一个台阶。本教材力求通俗易懂，便于自学，同时也考虑到使教师在教学中易于拓宽。书中配有一定数量的习题，与之配套的《C 语言程序设计实验与习题》一书中对其中大多数的习题给出了参考答案，并附加了大量的习题及其参考答案。

本书主要针对计算机语言的初学者编写，适用于各类院校非计算机专业本、专科学生，也可供高等职业技术学院、网络学院、成教学院学生，计算机等级考试者，以及培训班学员、C 语言自学者学习使用。

本书由马德骏、张建宏、汤练兵主编，第一、二、十一章由张建宏编写，第三、九、十章由汤练兵编写，第四、五、六、七、八章由马德骏编写。在制定本书的编写大纲中，傅世海、黄启荃、陈志铭、汤英、杨朝阳等同志给予了大力支持，提出了许多宝贵的意见，在此表示诚挚的谢意！

碍于我们的水平和时间，书中难免存在不少缺点和错误，敬请读者和同行专家不吝赐教。

编者

2002 年 10 月

目　录

第一章 C语言程序设计基础知识

1.1 概 述

1.1.1 信息的表示

1. 数制

在日常生活中，十进制是大家习惯使用的数制，而在电子计算机中主要使用二进制，这是由电子计算机的物理特性所决定的。二进制具有数符少、容易表示、运算简单可靠、便于物理实现、节省设备等优点，所以计算机数据几乎全是采用二进制数表示。

(1) 十进制数。对十进制数我们比较熟悉，它有 0,1,2,3,…,9 这十个数字符号，进行加、减法时，遵循“逢十进一，借一当十”的规则。在一个十进制数中，数字(又称数码)所在位置(数位)不同，则所代表的数值也不同。不同数位上单位数值的大小称为该数位的权值(又称位权)，例如 345 中的 3 处于百位上，权值为 10^2，所以可用 3×10^2 来表示。以此类推，我们可以将 234.56写成以 10 为底的幂指数按权展开式，即

$$234.56=2\times10^2+3\times10^1+4\times10^0+5\times10^{-1}+6\times10^{-2}$$

一般地，任意一个有 n 位整数和 m 位小数的十进制数 N 可以表示为

$$(N)_{10}=(k_{n-1}k_{n-2}k_{n-3}\cdots k_1k_0k_{-1}k_{-2}\cdots k_{-m})_{10}=\sum_{i=n-1}^{-m}k_i\times10^i \tag{1-1}$$

进一步，若将上述公式中的 10 换为 R(R≥2 的整数)，则此公式可推广到任意进制数值系统当中去，即

$$N=\sum_{i=n-1}^{-m}k_i\times R^i \tag{1-2}$$

这里 R 为基数，代表某一种数制，R^i 为第 i 位的权值，十进制时 R=10。k_i 值只能是 0 到 R−1这些数字符号中的一个，如十进制时，k_i 的取值为 0 到 9。

(2) 二进制数。二进制与十进制一样，也属于进位计数制。二进制仅有 0 和 1 两个数字符号，权值为 2^i。例如，二进制数$(1110.1)_2$可以按公式(1-2)展开表示为 $1\times2^3+1\times2^2+1\times2^1+0\times2^0+1\times2^{-1}$。在进行加法、减法运算时，二进制遵循“逢二进一，借一当二”的规则。例如，对于$(1110.1)_2+(1100.1)_2$ 和$(1110.1)_2-(1101.1)_2$，如同十进制加减法运算一样：首先，小数点对位；而后，逐位相加或相减。要注意的是，要遵循二进制的规则。

$$\begin{array}{r} 1110.1 \\ +1100.1 \\ \hline 1\ 1011.0 \end{array} \qquad \begin{array}{r} 1110.1 \\ -1101.1 \\ \hline 0001.0 \end{array}$$

至于二进制乘法和除法，与十进制乘除法类似，这里不再叙述。

(3) 八进制数。八进制有 0,1,2,3,4,5,6,7 八个数字符号，权值为 8^i。例如，八进制数$(203.1)_8$可以按公式(1-2)展开表示为 $2\times8^2+0\times8^1+3\times8^0+1\times8^{-1}$。在进行加法、减法运算时，八进制遵循“逢八进一，借一当八”的规则。

(4) 十六进制数。十六进制有 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F 十六个数字符号，

权值为 16^i。例如，十六进制数$(2A3.1)_{16}$可以按公式(1-2)展开表示为 $2\times16^2+A\times16^1+3\times16^0+1\times16^{-1}$。注意，在十六进制中，A 到 F 这五个数字符号分别代表的十进制数值是的 10 到 15。在进行加法、减法运算时，十六进制遵循"逢十六进一，借一当十六"的规则。

(5) 不同数制间的转换

1) 二进制、八进制、十六进制数转换为十进制数

转换时，通常按位权法进行，即按十进制运算法则，将各位上的数码乘以该位的权值再进行累加所得到的数就是相应的十进制数。如：

$$(1110.01)_2=1\times2^3+1\times2^2+1\times2^1+0\times2^0+0\times2^{-1}+1\times2^{-2}=(14.25)_{10}$$

$$(125.11)_8=1\times8^2+2\times8^1+5\times8^0+1\times8^{-1}+1\times8^{-2}=(85.140625)_{10}$$

$$(B1.2)_{16}=11\times16^1+1\times16^0+2\times16^{-1}=(177.125)_{10}$$

2) 二进制、八进制、十六进制之间的转换

· 二进制数转换成八进制或十六进制数：

因为二进制、八进制、十六进制之间有内在的对应关系：$2^3=8$，$2^4=16$；3 个二进制位所能表示的最大整数为 7，最小非负整数为 0；4 个二进制位则相应为 15 和 0。由于有这种对应关系的存在，因此在将一个二进制数转换为八进制时，对其整数部分从小数点开始向左三位一组，进行划分，高位不够三位时，左边补 0；对其小数部分从小数点开始向右三位一组，低位不够三位时，右边补 0；然后，再将各组二进制数分别转换成八进制数即可。将一个二进制数转换为十六进制数时，也采用上述方法，只是分组时是 4 个二进制位一组即可。如：

$$(1010111.01111)_2=(001\ 010\ 111.011\ 110)_2=(127.36)_8$$

$$(1011111.01111)_2=(0101\ 1111.0111\ 1000)_2=(5F.78)_{16}$$

· 八进制数、十六进制数转换成二进制数：

方法很简单：对于八进制数，只需将每个数字用 3 个二进制数位来表示；对于十六进制数，用 4 个二进制数位来表示即可。如：

$$(32.12)_8=(011\ 010\ .001\ 010)_2=(011010.001010)_2$$

$$(9A.0C)_{16}=(1001\ 1010\ .0000\ 1100)_2=(10011010.00001100)_2$$

· 八进制数、十六进制数间的转换：

通常，它们之间的转换是利用二进制数作为过渡进行变换的，即先将一种数据转换成二进制数，再将该二进制数转换成另一种数据。如：

$$(A5)_{16}=(1010\ 0101)_2=(010\ 100\ 101)_2=(245)_8$$

3) 十进制数转换成二进制、八进制、十六进制数

· 十进制数转换成二进制数：

对于十进制数的整数部分采用"除二取余"的方法，即：将该十进制数整数部分除以 2，所得余数作为二进制数整数部分的最低位，再将商作为被除数除以 2，所得余数作为二进制数的次低位，依次类推，可以的到一系列余数，按其先后从右向左排列，就构成了二进制数的整数部分。对于小数部分，采用"乘二取整"的方法，即：将该十进制数小数部分乘以 2，取积的整数部分，作为二进制数小数点后的第一位，再将积的小数部分乘以 2，取其整数部分，作为二进制数小数点后第二位，依次类推，就可得到二进制数的小数部分。将整数部分和小数部分合并，即得到对应的二进制数。如：求$(28.125)_{10}$所对应的二进制数，处理过程如下：

余数

2 | 28 …… 0 ……最低位
2 | 14 …… 0
2 | 7 …… 1
2 | 3 …… 1
1 ……最高位

0.125×2=0.25 ……0 最高位
0.25×2=0.5 ……0
0.5×2=1.0 ……1 最低位

整数部分：$(28)_{10}=(11100)_2$　　小数部分：$(0.125)_{10}=(0.001)_2$

合并后得到：　$(28.125)_{10}=(11100.001)_2$

· 十进制数转换成八进制、十六进制数：

原则上与十进制数转换成二进制数的方法一样，只是进行除法运算时是除 8 或 16 取余，进行乘法运算时是乘 8 或 16 取整。建议实际转换时，先转换成二进制数，再由二进制转换成八进制或十六进制数，这样做简单一些。如：求$(10.25)_{10}$所对应的八进制和十六进制，过程如下：

$(10.25)_{10}=(1010.01)_2=(001\ 010.010)_2=(12.2)_8=(1010.0100)_2=(A.4)_{16}$

2. 数据的编码

计算机中的数据分为两大类：数值型数据和非数值型数据。数值型数据是指可以进行算术运算的数据，如$(256)_{10}$，$(0111.101)_2$等都是数值型数据。非数值型数据通常不参加算术运算，如字符串“你好，世界”，“2002.6.15”等，就是典型的非数值型数据。

数值型数据的表示

在计算机中表示一个数值型数据，首先需要解决数的长度、符号和小数点位置的表示等问题。通常，在计算机中，使用固定长度的二进制位数来表示数值型数据，如：8 个、16 个、32 个二进制位来表示一个数值型数据。在计算机中，数值型数据的小数点位置总是隐含的，以便节省存储空间。隐含的小数点位置可以是固定的，也可以是可变的。前者称为定点数(fixed point number)，后者称为浮点数(floating point number)。

(1) 定点数。在计算机中，整数是按定点数格式存放的。

有符(signed)定点整数格式如图 1-1 所示。其中最左边的一位二进制位是符号位，若该位为 0，则表示该数为正数；若为 1，则表示为负数。其余为数值部分，小数点隐含在最右边。

无符(unsigned)定点整数格式如图 1-2 所示。其中全部二进制位用来表示数值部分，小数点隐含在最右边。

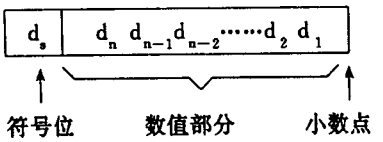


图 1-1　有符定点整数格式

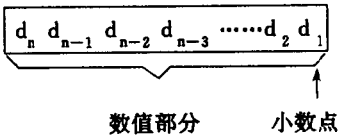


图 1-2　无符定点整数格式

(2) 浮点数。在计算机中，浮点数的格式如图 1-3 所示。其中浮点数由阶码和尾数两部分组成，阶码是整数，阶符和阶码的位数 n 合起来确定了浮点数的表示范围以及小数点的位置；尾数是小数，其位数 m 确定浮点数的精度(有效数字的位数)；尾数的符号 d_s 是浮点数的符号。

在解决了上述问题以后，还有一个值得考虑的就是数值型数据的编码问题。常用的编码方

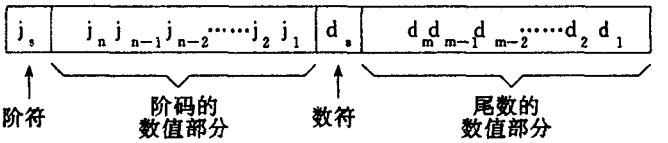


图 1-3 浮点数的格式

法有原码、反码和补码方法，采用何种编码方法对于计算机的运算效率有很大影响。

(3) 原码表示法。假定用八位二进制位来表示一个二进制整数 a 的原码，则：最高位为符号位，0 表示正数，1 表示负数，其余的二进制位用来表示 a 的绝对值。如：

$a_{真值}=(+51)_{10}=(+00110011)_2$　　则 $[a]_{原}=00110011$

$a_{真值}=(-127)_{10}=(-01111111)_2$　　则 $[a]_{原}=11111111$

原码表示法在八位二进制位中表示范围如下：

$-127\leqslant a\leqslant+127$　　$[+0]_{原}=00000000$　　$[-0]_{原}=10000000$

(4) 反码表示法。我们可以通过一个二进制整数的原码得到所谓反码。假定用八位二进制位来表示一个二进制整数 a 的反码，则：当 a 为正数时，$[a]_{原}=[a]_{反}$；当 a 为负数时，则保持 $[a]_{原}$ 的符号位不变，其余各个二进制位逐位取反，即 0 变 1，1 变 0。如：

若 $[a]_{原}=00110011$　则　$[a]_{反}=00110011$

若 $[a]_{原}=10110011$　则　$[a]_{反}=11001100$

反码表示法在八位二进制位中表示范围如下：

$-127\leqslant a\leqslant+127$　　$[+0]_{反}=00000000$　　$[-0]_{反}=11111111$

(5) 补码表示法。同样，我们可以通过一个二进制整数的反码得到其补码，即：当 $a\geqslant0$ 时，$[a]_{补}=[a]_{反}$；当 $a<0$ 时，则 $[a]_{补}=[a]_{反}+1$。如：

若 $[a]_{反}=00110011$　则　$[a]_{补}=00110011$

若 $[a]_{反}=10110011$　则　$[a]_{补}=10110100$

补码表示法在八位二进制位中表示范围如下：

$-128\leqslant a\leqslant+127$　　$[0]_{补}=00000000$　　$[-128]_{补}=10000000$

应当注意到，在补码表示法中，0 的表示是惟一的，而最小值可达 −128。在上述介绍当中我们是以八位二进制位为例，同样，我们可以将其推广到十六位，三十二位等二进制位中去，这里不再叙述。

(6) 补码的加、减法。补码的加法公式是：

$$[a]_{补}+[b]_{补}=[a+b]_{补}$$

补码的减法公式是：

$$[a-b]_{补}=[a]_{补}-[b]_{补}=[a]_{补}+[-b]_{补}$$

例 1.1　$a=[11]_{10}=[00001011]_{原}$，$b=[5]_{10}=[00000101]_{原}$，求 a+b。

因为 $[a]_{补}=00001011$，$[b]_{补}=00000101$，故

$$\begin{array}{rl} & 00001011 \\ \text{补码}\rightarrow & \underline{+00000101} \\ & 00010000 \end{array}$$

$[a+b]_{补}=[a]_{补}+[b]_{补}=[00010000]_{补}$

$[a+b]_{反}=[00010000]_{反}$

$$[a+b]_{原}=[00010000]_{原}=[16]_{10}$$

例 1.2 $-a=[-11]_{10}=[10001011]_{原}$，$b=[5]_{10}=[00000101]_{原}$，求 b−a。

因为$[-a]_{补}=11110101$，$[b]_{补}=00000101$，故

$$\begin{array}{rr} & 00000101 \\ \text{补码}\rightarrow & +11110101 \\ \hline & 11111010 \end{array}$$

$$[b-a]_{补}=[b]_{补}+[-a]_{补}=[11111010]_{补}$$

$$[b-a]_{反}=[11111001]_{反}$$

$$[b-a]_{原}=[10000110]_{原}=[-6]_{10}$$

在 Turbo C 中，有符整数是按有符定点整数格式表示的，无符整数是按无符定点整数格式表示的。

字符型数据的表示

在计算机所使用的数据中，还有另一种类型的数据，即字符型数据，它包括字母、文字、符号、数字等。由于计算机内部所有的信息都是以二进制形式存放的，所以必须按照某种规则对字符数据进行处理。对于任意一个计算机可以识别的字符数据，都按照特定编码规则使其与一个二进制编码建立一一对应的关系，也就是说在这种编码规则下，用一个二进制编码表示一个字符数据。对于各类字符有不同的编码规则，如对于英文字符、符号等字符有 ASCII 码、BCD 码，对于中文有 GB2312 国标码等各种不同的编码规则，而且这些编码规则一般是国家标准或国际标准，为国家或国际上所承认并且执行的。

(1) ASCII 码。ASCII 码是计算机系统中广泛使用的一种字符编码，是英文 American Standard Code for Information Interchange(美国信息交换标准编码)的缩写。该编码已经被国际标准化组织所采纳，成为国际间通用的信息交换标准编码。目前国际上流行的是 ASCII 编码的七位版本，即用一个字节的低七位表示一个字符，高位充零。七个二进制位可表示 128 种状态，故可用来表示 128 个不同的字符，在 ASCII 编码的七位版本中表示 33 个通用控制字符、95 个可打印显示的字符(其中 10 个数字、52 个大小写英文字母、33 个标点符号和运算符号)。ASCII 码字符表见附录 I。

(2) 国家标准汉字编码。GB2312-80(国家标准汉字编码)是常用的汉字编码标准，它收录了 6763 个常用汉字，同时根据这些汉字使用频率的高低，又将它们分成两部分：一部分称为一级汉字共 3755 个，即最常用的汉字；另一部分称为二级汉字共 3008 个，为次常用的汉字。GB2312-80 还收录了一些数字符号、图形符号、外文字母等。

GB2312-80 规定用连续的两个字节来表示一个汉字，并且只用各个字节的低 7 位，最高位未定义，这样以来就有可能与 ASCII 码字符产生冲突。就单个字节来说，两种编码方式都只用到字节的低七位，ASCII 码规定高位充零，而国标码对高位未定义，因此，对单个字节而言，不能确定它到底是一个 ASCII 码字符还是一个汉字的一部分(低字节或高字节)。于是有很多为了解决这类问题的方案应运而生，变形国标码就是其中之一，并且得到了广泛的应用。它的主要特点是将国标码编码的各个字节的最高位置 1，以达到区别于 ASCII 编码的目的。

由于计算机中各种信息都是以二进制形式存在，有的是数值，有的是 ASCII 码字符，有的是汉字，如何区分它们呢？这实际上取决于我们(或者程序)按照何种规则判读它们，例如：对于机器内存中连续两个字节，它们的低七位内容分别为 0110000 和 0100001，如果它们的最高位均为 1，则表示汉字“啊”；如果均为 0，则表示为两个 ASCII 码字符“0”和“!”。当然，我们还可

根据不同的数值编码规则将它们判读成不同的数值，这里不再详细叙述。

1.1.2 计算机系统的基本组成

一个完整的计算机系统是由硬件系统和软件系统两大部分组成。硬件系统是构成计算机系统的各种物理设备的总称，是计算机系统的物质基础，它由运算器、控制器、存储器、输入设备和输出设备组成；软件系统是为运行、管理和维护计算机而编制的程序和各种文档的总和。

计算机系统的组成如图 1-4 所示。

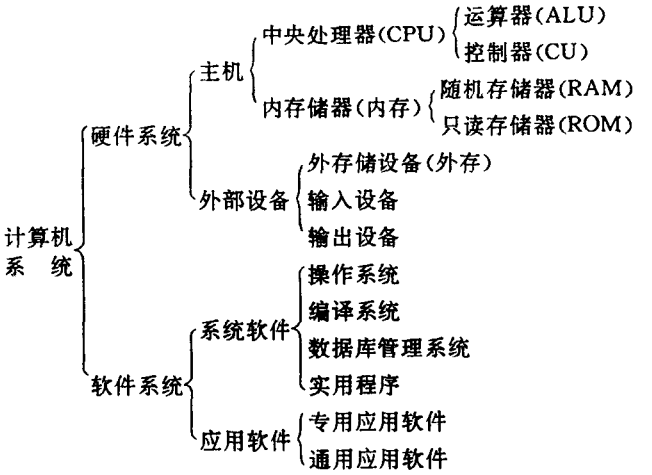


图 1-4 计算机系统的组成

1. 硬件系统

计算机硬件主要由中央处理器、主存储器、辅助存储器、输入设备、输出设备等组成。

(1) 中央处理器(CPU)。CPU 的作用是不断地从内存中取指令并执行指令。由机器码组成的这种指令序列，称为 CPU 的可执行程序。CPU 严格按照程序(即由内存中取来的指令序列)工作，CPU 的指令控制部件负责解释并执行各条指令，在内部进行算术运算、逻辑运算，对外部按指令要求与内存、接口设备交换信息，协调整个计算机系统的工作。

(2) 主存储器(main memory)。主存储器又称为内存储器，简称内存，用来存放当前运行的程序和数据。根据其工作方式和读写功能的不同，可分为：只读存储器和随机存储器。

1) 只读存储器(ROM)。ROM(Read Only Memory)中存放的信息在一般情况下只能被读出，不能被改写，通常用来存放那些永久不变的信息，如计算机加电自检程序、启动引导程序等。它的内容是由厂家出厂前写入的，一旦写好，用户就不能随便更改。

2) 随机存储器(RAM)。与 ROM 不同，RAM(Random Access Memory)中的信息有以下两个特点：一是随机存取信息，即信息不仅可以随时读取，而且还可以随时写入；二是具有“挥发性”，当电源关闭后，保存的信息会消失，且不可恢复。

(3) 辅助存储器(auxiliary memory)。主存储器速度快但价格昂贵，因而容量受到限制，且 RAM 中的信息不能长期保存，断电后即“挥发”，所以计算机采用了大容量的辅助存储器，如磁带、磁盘、光盘等。辅助存储器只能与主存储器交换信息，是主存储器的扩充，它与主存储器一起构成计算机存储体系中的一个重要组成部分。

在存储器中，用来存放一个二进制代码位的存储器最小单位是存储位(bit)，由若干个存储位组成一个存储单元，如 8 个二进制位组成一个字节(byte)存储单元，然后再由许多个存储

单元组成存储器。为了区分不同的存储单元，必须将它们逐一编号，该编号称为存储单元的地址。通过地址，可以访问各个存储单元。一个存储器中包含的存储单元的总数通常称为该存储器的存储容量。容量越大，能存储的信息越多。存储容量常用字节数来表示，如 64KB、128MB 等。存储容量的换算关系如下：

1byte＝8bits

1KB＝1024bytes＝2^{10}bytes　　1MB＝1024KB＝2^{20}bytes

1GB＝1024MB＝2^{30}bytes　　1TB＝1024GB＝2^{40}bytes

（4）输入/输出设备。输入设备是向计算机输入数据、信息的设备的总称。它将计算机程序、文本信息、多媒体信息以及各种数据转换成计算机能处理的数据形式并输送到计算机。常见的输入设备有键盘、鼠标、扫描仪等。输出设备是能将计算机处理好的信息转换成文本、图形、多媒体等形式并输出的设备。常见的输出设备有显示器、打印机、绘图仪等。

2. 软件系统

计算机只有配备了软件系统才能进行工作。一台计算机能否发挥其应有的作用，实现硬件系统所能完成的信息处理功能，取决于软件系统的优良与否。

软件一般指计算机运行所需的各种程序、数据以及相关的文档。软件系统由系统软件和应用软件两大部分组成。系统软件是用来对计算机进行管理、控制和维护，以及支持应用程序运行的软件的集合。应用软件是在系统软件的支持下为解决各类实际问题而设计开发的软件（程序）。

（1）系统软件。系统软件用于管理计算机资源，分配和协调计算机各部件工作，提高计算机的使用效率，方便用户使用计算机。系统软件包括以下四大类：

1）操作系统：是用于管理计算机系统资源的程序的集合。它的主要功能是处理机管理、存储器管理、文件管理、设备管理、作业管理。常见的操作系统有 DOS，Windows，Unix 等。操作系统是计算机和用户之间的接口。

2）编译系统：是指各种程序设计语言机器处理程序。计算机程序设计语言分为低级语言和高级语言，低级语言包括机器语言、汇编语言，高级语言包括 Basic，Fortran，C 等。

3）实用程序：为系统维护和运行提供便捷而有效的服务性程序。这类程序称为实用程序，它的种类很多，通常包括诊断程序、调试程序、文本编辑程序等。

4）数据库管理系统：是实现有组织地、动态地存储大量相关数据，方便用户和应用程序访问的计算机软件、硬件资源组成的系统。它包括数据库和数据库管理软件，用于对大量数据的存储、整理、查询、维护和各种报表的建立以及数据资源的多用户、多应用程序的共享等。常用的数据库管理系统有 Foxpro，Oracle 等。

（2）应用软件。应用软件是用户或软件开发人员在系统软件的支持下，为解决各类实际问题而设计、开发的软件，它包括通用应用软件（即软件包）和专用应用软件（即用户应用软件）。通用应用软件是指由软件公司专业人员为解决通用性问题而设计的软件，以供用户选择使用。这类软件很多，如 Office 2000（办公自动化），Access 97（数据库），SAS（统计分析系统）等。专用应用软件是指用户为了解决特定问题，自己或委托他人研制开发的软件，如工资管理系统等。

3. 计算机语言

计算机语言是人类为了有效地与计算机进行信息的传递、沟通，并且使计算机能够按照人类的意志进行工作而开发出的一种语言。人类使用它描述解决问题的一系列步骤，计算机能够

识别并执行它，以达到解决问题的目的。

(1) 机器语言。计算机能够理解并执行的有“0”和“1”组成的二进制指令，即机器指令。机器语言就是这样的机器指令的集合。用机器指令所编写的解决某一实际问题的一系列指令就称为机器语言程序。显然，一方面由于机器指令由“0”和“1”组成，用它编制出来的程序机器可以直接识别并执行，执行的效率比较高；但另一方面，正是由于机器指令的难记、难写、难懂、难调试、难移植等特点，使得用机器语言编制程序成为一件非常冗长、繁琐、甚至是“痛苦”的工作。在计算机产生初期，人们只能使用机器语言编制程序。

(2) 汇编语言。为了克服机器语言的缺点，计算机工作者采用了助记符的方法，使每条机器指令与一个符号一一对应，便于记忆和书写。例如用 ADD 表示加法操作，用 SUB 表示减法操作等。这些特定的符号和一些相应的格式和规则，构成了所谓的汇编语言。

汇编语言是面向机器的语言，但机器并不能直接执行，需要使用“编译”程序将它翻译成为机器指令后，计算机方可执行。

(3) 高级语言。虽然用汇编语言编写程序比用机器语言来得方便些，但是仍然很“麻烦”，除了要记众多的助记符和规则外，还需要了解有关的计算机硬件知识，这对于一般计算机用户是一件困难的事情。于是计算机工作者设计出面向应用的计算机语言，即高级语言，它更接近人类的自然语言，便于学习、理解和使用。

显而易见的是，用高级语言编写的程序也不能直接地被计算机执行，只有经过“编译”或“解释”程序处理后才能在计算机上执行。

4. 翻译方式

在上面的叙述当中，我们多次提及使用编译程序或解释程序对高级语言程序(或汇编语言程序)进行“翻译”处理，使它成为机器指令程序，才能提交计算机执行。这里需注意：其核心是将高级语言(或汇编语言)转换成“机器指令程序”。下面分别介绍这两种处理方式。

(1) 解释方式。使用解释程序(interpreter)将高级语言程序的语句逐条“翻译”成机器指令并逐条提交计算机执行，直至程序结束(如图 1-5 所示)，当高级语言程序中的语句出现错误时立即终止执行，并给出错误信息以便修改后重新解释执行。

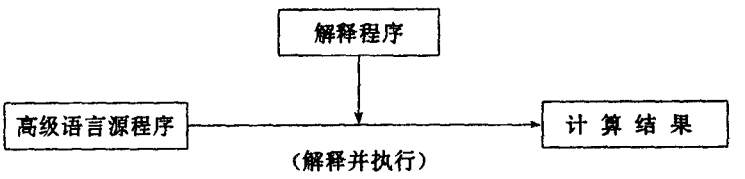


图 1-5　解释方式

(2) 编译方式。分两步进行：首先通过编译程序(compiler)对整个高级语言程序进行编译，它包括翻译和查错(词法分析、语法和语义分析、生成和优化目标程序)，出现错误时，停止编译，报告错误，不生成目标程序，待修改源程序后，再进行编译，直到最终得到正确的目标程序；然后使用链接程序(linker)对目标程序进行链接，得到可执行的程序，这时才能将可执行程序提交计算机执行(如图 1-6 所示)。

解释方式占用内存空间少，但运行效率低；编译方式占用内存空间多，运行效率高。我们将要学习的 Turbo C 是以编译方式工作的。