

周建中

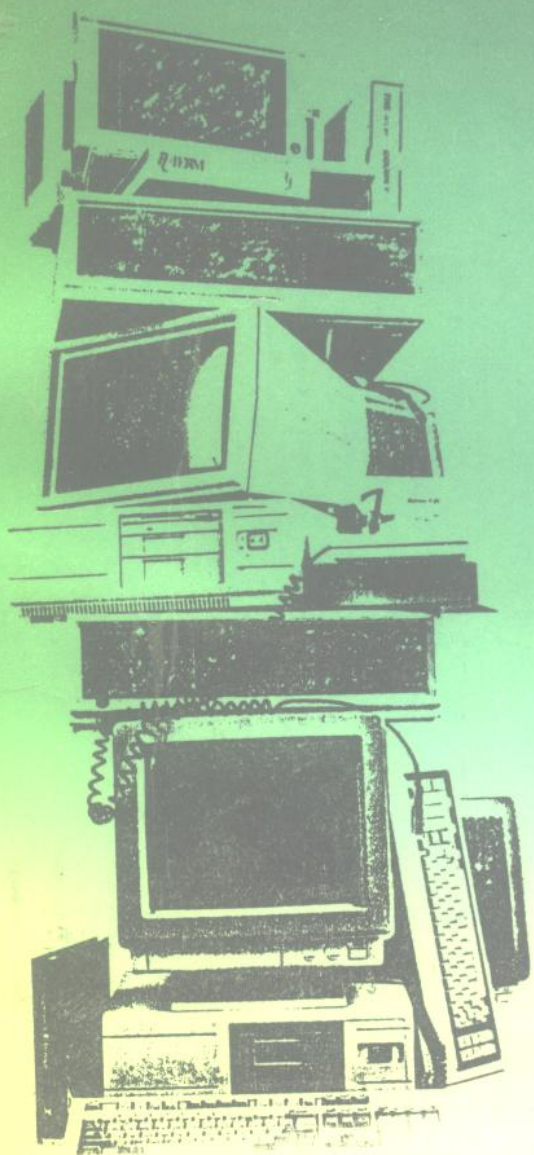
陈强璋

杨宗源

宁鲁生

编

C语言程序设计





C语言程序设计



周建中 陈强璋 杨宗源 宁鲁生 编

华东师范大学出版社



(沪)新登字第201号

C语言程序设计

周建中 陈强璋 杨宗源 宁鲁生 编

华东师范大学出版社出版发行

(上海中山北路3663号)

邮政编码: 200062

新华书店上海发行所经销 浙江大学印刷厂印刷

开本: 850×1168 1/32 印张: 11.25 插页: 1 字数: 289千字

1994年7月第一版

1994年7月第一次印刷

印数: 001—5,000本

ISBN7-5617-1102-6/H·094

定价: 8.70元

前 言

C语言是一种高效、实用的程序设计语言。在C语言诞生初期，即被用来书写UNIX操作系统。由于UNIX操作系统本身的合理性以及C语言的高度可移植性，使UNIX操作系统广为流行，各种计算机系统中几乎都配置了UNIX操作系统。使用C语言书写UNIX操作系统，被公认为是用高级程序设计语言书写系统软件的成功典范。不仅如此，在C语言问世至今的二十余年中，C语言还被用来书写各种应用软件和支援软件。从网络软件、数据库管理系统和各种图形软件，到各种特定应用领域中的专用软件，C语言的使用几乎处处可见，它已成为人们广泛接受并普遍使用的一种程序设计语言。

不仅是计算机专业人员广泛使用C语言，计算机应用人员也在越来越多地学习、使用C语言。近年来，不少高校的计算机专业和各类非计算机专业都开设了C语言课程，而且，C语言被指定为计算机专业人员的水平考试和非计算机专业学生的等级考试中可选用的程序设计语言之一，这无疑将进一步提高人们学习、使用C语言的兴趣。

本书是为非计算机专业的C语言课程编写的一本教材，内容以目前所发表的ANSI C语言标准为基准，其范围与深度与“上海市高等学校非计算机专业学生计算机等级考试”中有关C语言的考试大纲相一致。

在本书的编写过程中，考虑到读者的对象基本上是非计算机专业人员，因此，叙述上力求通俗易懂；在C语言学习过程中的重点和难点所在的部分，尽量叙述充分，并配以帮助理解 and 使用的实例，以便使读者能自行掌握重点并克服难点。

本书共分十二章。第一章为读者提供基本的计算机系统组成

框架的概念，C语言与硬件计算机的联系较为紧密，了解这些概念对于学习C语言来说是必要的。除此之外，还介绍了算法和程序设计语言有关的某些基本概念。第二章为读者提供C语言源程序组成方面的概貌。第三章介绍C语言的基本数据类型。为使读者能充分理解以后若干章节中的实例，本章还介绍了常用的输入、输出，以及广泛使用的数据类型——数组。第四章介绍C语言提供的各类基本运算和表达式。第五章介绍C语言的各种语句。第六章介绍C语言的函数定义及函数调用。第七章介绍各种变量的存贮类型。第八章介绍在基本数据类型上构造新类型的方法。第九章深入讨论指针这种可能使程序的效率得以提高的数据类型。第十章简要地介绍了C语言的各种预处理功能。第十一章介绍各种常用的库函数。第十二章为读者介绍结构化程序设计方法的初步知识。遵循本章提出的基本原则，将使设计的程序有清晰的逻辑结构，使人易于理解。附录中介绍了TURBO C上机操作的基本步骤及过程。本书中的所有实例均可在TURBO C(2.0版)中运行。

本书第一、十、十二章由宁鲁生编写，第二、六、七章由周建中编写，第三、八、九章由杨宗源编写，第四、五、十一章由陈强璋编写。华东师范大学出版社为本书出版做了大量工作，在此表示感谢。

限于编者水平，疏漏之处在所难免，恳请读者批评指正。

编 者

目 录

第一章	计算机系统、算法和程序设计语言	(1)
§1.1	作为解题工具的计算机系统	(1)
§1.2	算法	(6)
§1.3	程序设计与程序设计语言	(19)
第二章	C 语言基本知识和源程序结构	(24)
§2.1	C 语言的历史和基本特点	(24)
§2.2	C 语言源程序结构	(27)
§2.3	一个简单的程序实例	(32)
第三章	数据及类型	(38)
§3.1	常量	(38)
§3.2	变量及类型	(42)
§3.3	指针变量	(44)
§3.4	简单的输入、输出	(47)
§3.5	数组	(49)
第四章	基本运算和表达式	(57)
§4.1	运算概述	(57)
§4.2	算术运算和关系运算	(58)
§4.3	逻辑、位逻辑和移位运算	(64)
§4.4	取地址和间接存取	(69)
§4.5	增减 1 和赋值运算	(70)
§4.6	其他运算符	(78)
§4.7	表达式	(82)
第五章	语句	(88)
§5.1	基本语句	(88)

§5.2	选择控制语句	(89)
§5.3	重复控制语句	(99)
§5.4	其它控制语句	(114)
第六章	函数	(124)
§6.1	函数定义	(127)
§6.2	函数调用	(134)
§6.3	递归函数	(148)
§6.4	综合实例	(157)
第七章	存贮类型	(166)
§7.1	源程序组织和标识符的作用域	(166)
§7.2	变量的存贮类型	(169)
§7.3	分程序的变量环境	(187)
第八章	结构、联合及类型命名	(194)
§8.1	结构	(194)
§8.2	位域	(204)
§8.3	联合	(208)
§8.4	类型命名	(211)
第九章	指针	(216)
§9.1	指针与数组	(216)
§9.2	指针与结构	(227)
§9.3	指针与函数	(235)
§9.4	多级指针与类型说明	(251)
第十章	预处理命令	(259)
第十一章	库函数	(271)
§11.1	C 语言的库函数	(271)
§11.2	终端I/O库函数	(275)
§11.3	文件输入/输出库函数及文件操作	(283)

§11.4 其它库函数	(299)
第十二章 结构化程序设计初步	(309)
§12.1 结构化程序设计的目标	(310)
§12.2 结构化程序设计的基本原则	(312)
§12.3 结构化程序设计实例	(325)
附录A ASCII 字符代码表	(345)
附录B TURBO C 上机过程简介	(346)

第一章 计算机系统、算法和 程序设计语言

§1.1 作为解题工具的计算机系统

一、计算机发展简史

从人类学会使用金石、结绳计数开始，计算机的诞生经历了漫长的历史，直到1946年2月15日诞生了第一台被命名为埃尼阿克(ENIAC)的数字式电子计算机，开创了计算机技术发展的新时期——电子计算机时期。

从第一台电子计算机诞生至今，不过短短四十多年的时间，期间电子计算机的发展却已经历了四代。第一代是它的婴儿时期，其主要特征是：主机采用电子管器件，应用以科学计算为主，软件采用机器语言、符号语言编程。第二代的主要特征是：主机采用半导体器件，应用领域扩大到数据处理，软件采用高级语言编程并出现了诸如操作系统这样的系统软件。第三代计算机的主要特征是：主机采用集成电路器件，软件技术发展 to 操作系统成熟，软件工程兴起。应用领域进一步拓宽，终端设备迅速发展。第四代计算机的主要特征为：主机采用大规模集成电路器件，并向超大规模集成电路发展。应用领域扩大到社会生活的各行各业，以至于现代化社会离开计算机是不可想象的。计算机的发展已经逐步成为衡量社会文明程度的一种尺度。

二、计算机的基本结构

现代计算机系统是多种多样的，大、中、小和微型计算机在硬件配置上有很大差别。这里只介绍目前流行的计算机系统所共

有的基本配置。图1.1给出了计算机硬件系统的基本配置情况。

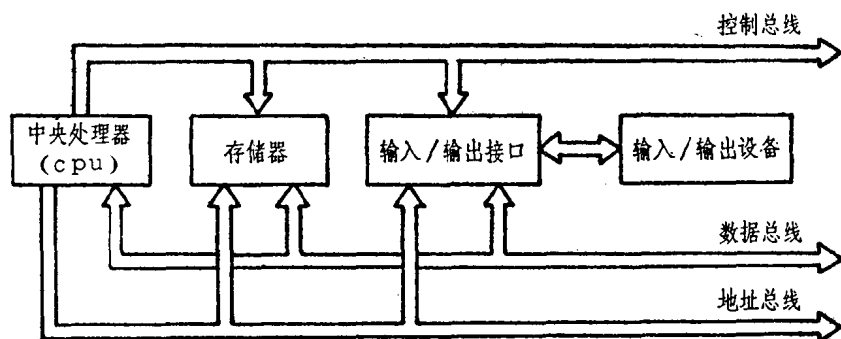


图 1.1 计算机硬件系统的基本配置

1. 中央处理器CPU(Central Processing Unit)

中央处理器是计算机硬件的核心部分，它与其他各部件不仅有地址信息与数据信息方面的交换联结，而且还有控制信息与状态信息方面的交换联结。由CPU发布控制信息到各个部件，控制它们动作。所以CPU是指挥硬件系统工作的首脑机构。中央处理器包括：

(1) 算术逻辑部件

算术逻辑部件的功能是实施各种算术和逻辑的运算，例如加、减、乘、除、移位和逻辑乘等运算。算术逻辑部件由加法器、进位线路和移位线路等构成。

(2) 数据寄存器

数据寄存器用于暂存即将参加某种运算操作的数据。例如寄存参与算术运算的数据、运算的中间结果以及某种控制用信息。

(3) 程序计数器

程序计数器又称指令地址计数器。其功能是存放当前要执行的指令地址。在按顺序逐条地执行指令时，每当执行完一条指

令，程序计数器就会自动进行加“1”操作。当要求下一条指令从某个地址开始时，可以强制程序计数器形成这个地址码，于是下一条指令就可以从新地址开始顺序执行下去。

(4) 地址寄存器

地址寄存器是存储器地址总线上的地址发源地。无论是可执行指令的地址还是指令中涉及到的操作数据的地址，都是经由地址寄存器发往地址总线的。

(5) 指令译码器

指令译码器的主要作用是把要执行的某一条指令的编码翻译成相应的操作控制信号，为执行并完成该指令的动作做好准备。

(6) 操作控制器

操作控制器的功能是根据指令译码器的输出去实际完成这条指令的动作。

2. 存储器(Memory)

存储器的主要功能是存放程序和数据。程序是计算机操作的依据，数据往往是机器操作的对象。计算机为了实现自动计算，把大量的信息预先存放在某个地方，这个地方就是存储器。

存储器通常由五个部分组成。

(1) 存储体部分

由贮存信息的介质组成，其主要功能是存储信息。存储体由一系列的存贮单元组成，每个存贮单元中可以存贮固定大小的信息，例如一条指令或一个数据。使用某种方法，例如用整数 0, 1, 2, ... 为每个存贮单元编号。指定编号之后，就可以从编号为 i 的存贮单元中取出指令或数据，或者把指令或数据存入该存贮单元中。存贮单元的编号称为地址，即在一个计算机系统中，每个存贮单元都有唯一确定的地址。

(2) 地址译码器部分

其功能是先对地址码进行翻译，然后找到与该地址码相对应

的那个存贮单元，为与那个存贮单元交换信息创造条件。

(3) 读写控制电路部分

其主要功能是确定交换信息的方式，是读出信息还是写入信息。

(4) 地址总线部分

其功能是为地址译码器提供地址码，地址译码器根据这个地址码去寻找相应的存贮单元。

(5) 数据总线部分

其功能是用来接纳或者发送被交换的信息。例如要存入某个单元的信息，都首先把它发送到数据总线上，然后将它写入相应的存贮单元中。

3. 输入输出设备

输入输出设备简称 I/O 设备，是人与计算机对话的设备。输入设备把人们要求计算机处理的数据、字符、文字、图形、图像、语言以及程序本身等各种形式的信息，转换成计算机所能接受的编码形式，送到计算机存入存储器内。输出设备则是把经过计算机处理后的结果（最终结果或中间结果）以人们所希望的形式表示出来。例如印刷、绘图、照相、语言等形式。

4. 输入输出接口

输入输出接口也称 I/O 接口，是 CPU 与 I/O 设备之间的接口。从图 1.1 中可以看到，I/O 设备通过 I/O 接口与 CPU 及存储器相连接。这是由于 I/O 设备往往含有相当数量的机械与电机、电气部分，其工作速度与 CPU 或存储器有很大差距，因此不可能直接匹配；另外在表达信息的格式上，两者也都有不同的要求，所以需要使用 I/O 接口来进行协调，以便使计算机系统能够按人们的要求来完成工作。

三、操作系统初步

使用现代计算机解题的操作者一般都是通过操作系统与计算

机打交道。例如调用一个文件，启用磁盘，显示数据，这些操作都是由操作系统来完成的。操作系统是一组控制程序的总称，它用来控制和管理计算机的各种资源，实现计算机系统运行自动化。因此，操作系统是系统软件中最核心的组成部分，它指挥整个计算机系统的各个部件，包括软件在内，协调一致地工作。操作系统的主要任务有两个，一是管理计算机系统的所有资源，如处理器、存储器、各种输入输出设备和各种软件；二是为计算机系统和用户之间提供接口，使用户不必掌握计算机及 I/O 设备的那些物理细节，而以方便、简洁的形式直接使用这些设备所提供的功能。

操作系统一般由以下几部分组成：

1. 处理机管理

为了能使 CPU 和速度不同的 I/O 设备并行工作，充分发挥计算机各部件的能力，操作系统引入了多道程序设计，即在一个时刻可以有几道作业同时在计算机内运行。在这种情况下，各作业都需要 CPU 供其运行，一般在一个计算机系统中，处理机的数目总是少于同时运行的作业数，这样就产生了各个作业竞争 CPU 的情况，产生了 CPU 的分配问题。操作系统的处理机管理机构就是为妥善管理处理机，满足各类作业的特殊要求而设计的。

2. 存贮管理

现代计算机系统至少有两级存储器，即内存和外存。所有数据在实际参加运算之前，都必须存贮在计算机的内存中，同样运行程序的代码也必须在运行开始前存贮在内存中。与外存相比，内存的容量通常要小些，但内存中存取数据的速度要比外存快得多；外存用于存放需要长期保存的信息，其特点是容量大，存取速度较慢，常做后备存储空间用。

操作系统的存贮管理机构主要实现对内存的管理，它包括以下三方面内容：

(1) 内存分配

内存分配的功能是,当几个作业同时存在于内存时,为各作业妥善合理地分配内存,以便使各作业所占用的存储区不发生冲突。

(2) 内存保护

内存保护的功能是保证各作业只能在分配给自己的存储区内活动,而不相互干扰。

(3) 内存扩充

由于多个作业同时占用内存时,内存容量往往显得不够用,有时甚至连一个作业在内存中都容纳不下,因此需要设法扩充内存,并且要求由系统承担这种扩充以便减轻用户负担。内存扩充机构就是为完成上述这种需要而设立的。

3. 文件系统

文件是存贮在存贮设备中的数据的一种组织形式。文件系统是负责存取和管理文件信息的软件机构。它是由目录和一组文件操作组成的。文件系统为用户提供了一种简便统一的存取和管理信息的方法。例如通过文件系统,我们可以用简单的命令查看文件的目录、查找一个文件、建立和删除一个文件等。

4. 输入输出设备管理

输入输出设备是计算机系统的一种资源。和处理机与内存一样,也必须对它们进行合理分配,以便满足各个作业的需要,提高I/O设备的利用率。同时输入输出设备管理机构也为用户提供了使用这些设备的简便方法。例如我们可以方便地进行屏幕打印、显示某个目录中的文件等操作,而不必顾及显示屏和打印机的细节。

§1.2 算 法

一、算法的基本概念

在日常工作和生活中,我们常常会遇到许多问题。为了能够

顺利解决这些问题，就必须采取不同的步骤，设计出各式各样的方案。通常我们把这种解决问题的步骤和方案称作解决问题的方法、技术和过程。在计算机科学中，把这种为解决某一类特定问题而设计的方法叫做算法。每当我们使用计算机进行解题时，就需要设计求解该题的算法。通俗地说，算法是一个有限条指令的集合，这些指令规定了解决某一问题的运算或操作的序列。

一个算法应该具有以下五个主要特征：

1. 确定性

算法中的每一条指令都必须清楚明了，没有二义性。也就是说每个要完成的动作在语义上不能存在模糊不清的成份。例如不允许出现“使变量 x 加3或者加4”这样的动作，而必须明确给出变量 x 到底是加3还是加4。

2. 可行性

每条指令都必须充分基本，使得操作可以在有限的时间内完成。例如实数的算术运算就不一定是可行的，因为通常可能产生无限长的十进制数。如果限定十进制数的位数，使实数在允许的误差范围内可用有限位十进制整数和小数表示，那么实数的算术运算是可行的。

3. 有限性

算法的有限性是指算法必须在执行有限条指令后能够终止，因此一个永不停止的算法实际上是没有意义的。

4. 输入

一个算法应该有 $m(m \geq 0)$ 个输入，该输入是算法所要求的原始数据。

5. 输出

一个算法应该有 $n(n \geq 1)$ 个输出，通常这些输出是算法对输入数据计算的结果。

下面我们来看一个兔子繁殖问题的例子。这个问题的描述

如下:

设每一对新生下来的兔子两个月之后开始生育一对兔子,以后将每月生育一对。依此类推,如果在年初时有一对新生兔子并且假定在这一年中兔子没有伤亡,试问到这年年底共有多少兔子?

分析题意,我们知道:(1)每一对新生的兔子必须在两个月后开始繁殖;(2)每一对兔子一旦开始生育,就每月繁殖一对;(3)年初时只有一对新生兔子。这就是说一月份有一对新生兔。由于这对兔子要两个月后开始繁殖,所以二月份仍然只有一对兔子。三月份时年初的那对兔子开始生育了一对兔子,加上原有的兔子本身,那么三月份就有了两对兔子。四月份时,二月份的一对兔子又繁殖了一对兔,而三月份的兔子数中包含原有的兔子数和还不能生育的新生兔子,所以四月份的兔子对数等于二月份兔子对数加上三月份的兔子对数。依此类推,我们得到了兔子繁殖的计算规律。即除了一月和二月之外,第 i 月的兔子对数等于第 $i-2$ 月和第 $i-1$ 月份的兔子对数之和。于是我们获得了求某月兔子对数的数学模型。

设 i 表示月份, F_i 为第 i 月份共有兔子的对数,则我们有:

$$F_i = \begin{cases} 1 & \text{当 } i \leq 2 \text{ 时;} \\ F_i = F_{i-2} + F_{i-1} & \text{当 } i > 2 \text{ 时.} \end{cases}$$

根据这个数学模型,我们可以求得一组兔子繁殖的数列1,1,2,3,5,8,13,21,34,55,89,144,233,...;这组数列是中世纪意大利著名数学家斐波那契(Fibonacci)也即比萨的莱翁那度在1202年写的一本名叫《算盘学》(Liber abaci)的书中由兔子的繁殖问题而引出的。后来人们把这组数叫做斐波那契数。

经过上面的详细分析,现在可以着手设计求解一年中兔子繁殖数量问题的算法了。

算法 1:

输入数据个数为 0;

输出: 年底兔子的总数;

算法步骤:

(1) 令变量 A 、 B 和 S 等于 1, $I = 2$;

(2) $I = I + 1$;

(3) 用 B 取代 A , 用 S 取代 B , 即 $B \rightarrow A$, $S \rightarrow B$;

(4) 令 $S = A + B$;

(5) 如果 $I < 12$ 转步骤(2)继续执行, 否则执行步骤(6);

(6) 打印 $S * 2$;

(7) 算法结束。

这个算法引入了 A 、 B 和 S 三个变量分别用来存放 F_{i-2} 、 F_{i-1} 和 F_i 的值。变量 I 表示月份。当 $I < 12$ 时算法不断循环, 分别求出 3 月份, 4 月份, 5 月份, ..., 12 月份的兔子对数。仅当 $I \geq 12$ 时, 算法才执行步骤(6)输出 12 月份时共有的兔子只数。注意算法中 S 的定义是存放当月的兔子对数, 所以输出的结果应该是表达式 $S * 2$ 的值。

纵观算法 1, 其中的每一步操作都十分明确可行且无二义性。而且算法当 $I \geq 12$ 时会自动停止并输出运算的结果。

二、算法的表示形式

在算法 1 中, 我们采用了大家熟悉的文字和数学符号来表述, 这就是算法的自然语言描述。用自然语言描述算法时, 往往显得叙述冗长, 有时还会有说不清楚的感觉, 这是因为自然语言固有的含糊性增加了算法描述的难度。尤其是在描述判断、分支和转移的过程中, 用文字描述就显得格外冗长、拖沓和繁琐, 使人读来不得要领, 各部分的逻辑关系也不容易理清。为此人们在大量的实践中创造了许多表示算法的方法。下面我们就介绍几种常用的、行之有效的方法。