C 语言程序设计

主　编　李秦伟
副主编　程小辉

重庆大学出版社

内 容 提 要

本书是按照电子信息工程本科专业的规划编写的，对数据类型与表达式、C 基本程序设计、分支程序、循环程序、函数、预处理、数组与字符串、指针、文件操作、程序分析与设计等内容进行了深入细致的讲解与阐述，同时溶入了编者与同事在 C 程序设计课程教学改革中的经验与教训，力求使读者学到扎实、实用的知识，养成良好的程序设计风格与习惯。

图书在版编目(CIP) 数据

C 语言程序设计/李秦伟主编 . —重庆: 重庆大学出版社, 2004. 8

(电子信息工程专业本科系列教材)

ISBN 7-5624-3162-0

Ⅰ. C. . .　Ⅱ. 李 . . .　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(2004) 第 048036 号

C 语言程序设计

主　编　李秦伟

副主编　程小辉

责任编辑: 彭　宁　王启志　王启敬　　版式设计: 彭　宁

责任校对: 何建云　　责任印制: 秦　梅

*

重庆大学出版社出版发行

出版人: 张鸽盛

社址: 重庆市沙坪坝正街 174 号重庆大学(A 区) 内

邮编: 400030

电话: (023) 65102378　65105781

传真: (023) 65103686　65105565

网址: http: / /www. cqup. com. cn

邮箱: fxk@cqup. com. cn(市场营销部)

全国新华书店经销

自贡新华印刷厂印刷

*

开本: 787× 1 092　1 /16　印张: 20. 75　字数: 518 千

2004 年 8 月第 1 版　　2004 年 8 月第 1 次印刷

印数: 1—4 000

ISBN 7-5624-3162-0 /TP · 486　定价: 29. 00 元

前言

C 族语言 C、C++ 和 JAVA 等是目前世界上最广泛使用的通用计算机程序设计语言，拥有广泛的用户群，C 语言甚至被认为是计算机专业程序员必须掌握的语言。C 语言是结构化语言，而且有许多不同于其他语言的特点：简洁紧凑，使用方便灵活，数据结构和运算符丰富，可以直接访问硬件，生成目标代码紧凑高效，良好的标准化程度和可移植性等。由于 C 的以上特点，C 和 C 族编程语言被广泛应用于各种系统软件、应用软件和组件的开发。由于 C 语言的广泛应用，新的 C 和 C 族语言编译系统不断出现，同时也使它成了首选的教学语言之一，被广泛使用在程序设计教学中。

本教材是按照电子信息工程本科专业的规划编写的。C 语言是面向程序员的语言，对程序员要求较高，程序员使用 C 语言编写程序会感到限制少、灵活性大，功能强，容易入门但全面掌握较难。本教材溶入了编者与同事在 C 程序设计课程教学改革中的经验与教训，在注重讲解基本原理、方法的基础上，力求使读者学到扎实、实用的知识，培养良好的程序设计风格和习惯。

本教材第 1 章重点介绍程序的概念与程序在计算机中的工作原理。第 2 ~11 章是传统的 C 语言的教学内容，第 2 章介绍了 C 语言语句的基本成分，第 3 章讲述顺序程序设计与控制台 I/O，第 4 ~5 章讲述算法的概念、分支与循环程序设计，第 6 章介绍函数设计和模块的概念，第 7 章是预处理，第 8 章介绍数组的概念及在数组上的操作，第 9 章介绍指针的概念、指针的应用和动态存储的概念，第 10 章重点讲述结构的概念与应用及联合、枚举、位域的概念，第 11 章介绍了流的概念和文件操作。第 12 章介绍了 C++ 语言和面向对象的概念，这是考虑到多数学校的电子信息工程专业不开设面向对象程序设计这门课程，而面向对象的设计思想是目前程序设计的主流。第 13 章从一个例子来分析如何设计一个完整的软件，目前多数学校的电子信息工程专业不开设软件工程课程，该章的目的是使读者建立软件工程的思想；而且从编者的教学经验看，完成一个 1 000 行左右的软件，对于学习 C 语言、提高实际编程能力非常有效。

本教材学时安排 54 ~70 学时（含课内上机 10 ~14 学时），C 语言是一门实践性很强的课程，建议再安排 30 学时左右的课外上机。课内学时具体安排如下：

章节	1	2	3	4	5	6	7	8	9	10	11	12	13
学时	2	4	4 +2	4 +2	4 +2	6 +2	2	5 +2	6 +2	6 +2	3	6	4

其中第 10 章中的枚举、位域与位运算、第 12 章、第 13 章为选讲内容，另外对第 9 章、第 10 章的其余内容、第 11 章的内容，也可根据实际情况酌情增减。

本教材由李秦伟编写第 1 章和第 7 ～13 章，徐效军编写第 2 章，保文星编写第 3 章和第 6 章，张向利编写第 4 ～5 章。全书由李秦伟、程小辉统稿。

本教材的例子全部在 Turbo C 和 Visual C++ 中调试通过，为了使例子简明扼要，在本教材的例子中基本没有考虑异常处理，例子程序中肯定存在着健壮性、可靠性、可扩展性、兼容性等质量问题。

由于编者水平有限，本教材中不可避免地存在着谬误和争议之处，诚恳地欢迎广大读者提出批评意见和建议。

编者

2004 年 2 月

目录

第 1 章 绪 论

1.1 计算机语言的发展

计算机能够进行各种工作，是因为计算机可以执行程序。什么是程序呢？在生活中学生每天按照下面顺序作息：起床洗漱→早点→上课→中餐→上课→晚餐→自习→就寝，这就是学生生活的程序。在计算机中，可以连续执行的一条条指令的集合（指令序列）就是程序，计算机是按照指令序列一条条地执行指令而工作的，程序的特点是时间性、有序性。要编写计算机程序，第一步是学会计算机程序设计语言。计算机程序设计语言是指能为计算机所接受的人与计算机交流信息的规则。计算机程序设计语言随程序技术的发展而发展。

计算机最早的语言是机器语言，机器指令的集合就是机器语言。由于不同型号的计算机其机器指令系统是不相同的，因此机器语言是“面向机器”的语言。用机器语言编写的程序，是软件的直接表达形式，可在机器上直接运行。例如，在 PC 机中下面两行二进制数字表示的指令实现了 6 和 11 的相加运算：

1011000000000110

0000010000001011

可见，机器语言有许多缺点：难理解、难记忆，编程工作量大，易出错，维护困难，移植性差，通用性很差等。

为了克服机器语言的缺点，人们设计了一些易于记忆的符号来代替机器指令。即用助记符代替机器指令中的操作码，用符号代替操作数的地址码，这就是符号语言，常称汇编语言。如上述 2 条机器指令可写成 2 条汇编指令：

```
MOV AL，6
ADD AL，11
```

显然，这比机器指令易理解和记忆。用汇编语言编写的程序，称为汇编源程序。它比机器语言更易理解、编制和修改。然而，计算机并不“认识”汇编源程序，必须经汇编程序将源程序转换为用机器语言表示的目标程序后，计算机才能执行。

汇编语言和机器语言都是面向机器的语言，只能在特定的机型上运行，它们编写的程序

运行效率高，代码短，可以直接操作和使用机器的资源。它们共同的缺点是：移植性差，与人的自然语言差别大，只有了解该型机器结构的程序员才能编写程序等。一般把机器语言和汇编语言都归为“低级语言”或“面向机器的语言”。

高级语言起始于 20 世纪 50 年代末期。一种语言的所谓高级、中级和低级并不是指它的功能强大、先进与否，这里所谓的高和低是指离硬件的近和远。高级语言与自然语言和数学语言更接近，用它们编写的程序可读性强，交流更方便，而且可移植性好。用一种高级语言写成的程序称为源程序，可以在具有该种语言编译系统的不同计算机上使用。源程序必须翻译成机器语言才能执行。逐条翻译并执行的翻译程序称为解释程序，例如 BASIC 语言解释程序。而将源程序一次翻译成目标程序然后再执行的翻译程序称为编译程序，例如 C 和C++ 编译程序。上述 6 和 11 的加法在 BASIC 语言和 C 语言中可以这样写：

BASIC 语言：
```
X = 6
X = X + 11
```
C 语言：
```
x = 6;
x = x + 11;
```
可见高级语言更容易被人理解和接受，当然越高级的语言也就越缺乏操作硬件的能力。20 世纪 60 年代中期，FORTRAN，LISP 和 ALGOL 已相继出现，而 BASIC 语言简单易学，曾经在我国风靡一时。早期的 BASIC 语言有一个特点：所有的变量都是全局的，也就是说在程序的任何一个地方都可以读写任何一个变量。这样的语言由于各程序段对变量的读写相互干扰，因而很难由多人合作开发出较大型的软件。

在 20 世纪 60 年代出现了结构化程序设计和局部变量。20 世纪 70 年代以来，随着结构化程序设计思想的提出和日益深入，这段时期问世的几种程序设计语言的控制结构大为简化，最具代表性的有 Pascal 和 C 语言。这些语言又称“面向过程的语言”。所谓“面向过程”，就是不必了解计算机的内部逻辑，而是主要集中在解题算法的逻辑和过程的描述上，使用过程语言编写程序，通过程序把解决问题的执行步骤告诉计算机。结构化程序设计在 20 世纪 80 年代达到了鼎盛时期。

在采用结构化程序设计的软件中，子函数和子过程是全局的。由于这个原因，在编写更大型的软件系统中出现了不少问题。为此，出现了将变量和操作处理变量的函数（或过程）封装在一起组成“对象”的程序设计方法，即“面向对象的程序设计”。在这种程序设计方法中，对象里的函数或子过程只在一个对象内有效，相当于是局部函数。在 20 世纪 90 年代，这种程序设计方法迅速发展起来。目前绝大部分编程工具都支持或部分支持面向对象的程序设计，其中最常见的面向对象的语言是 C++ 和 JAVA。

在面向对象的基础上，出现了“组件”编程，很典型的是 ActiveX 控件和 COM 组件，它们实现了跨语言平台的代码和资源共享，大大提高了软件开发的效率。

1.2 程序存储计算机的工作原理

目前使用的计算机是利用亚微米技术或纳米技术制造的超大规模集成电路计算机，但从第一台电子管计算机 ENIAC 至今，计算机最基本的原理没有发生根本性的变化。目前的计

算机大都是冯·洛依曼提出的三总线结构的程序存储计算机。

1.2.1　程序存储计算机的组成与工作原理

计算机系统是由硬件和软件两大部分组成，硬件主要有 5 个部分，它们是：控制器、运算器、存储器、输入设备和输出设备。硬件是软件的基础，软件是硬件的灵魂。计算机的组成如图 1.1 所示。

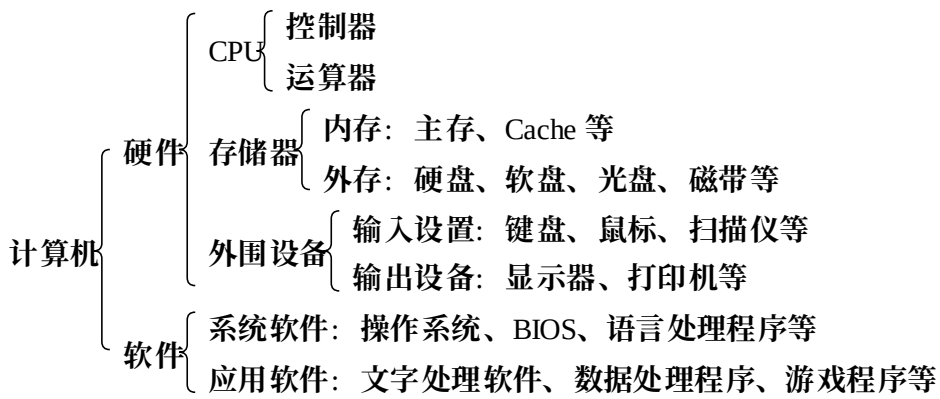


图 1.1　计算机的组成

在硬件的 5 个部分中，控制器和运算器统称为中央处理器，简称 CPU (Central Processing Unit)。为了提高计算机的速度，现在还把高速缓存 Cache 集成到了 CPU 内部。

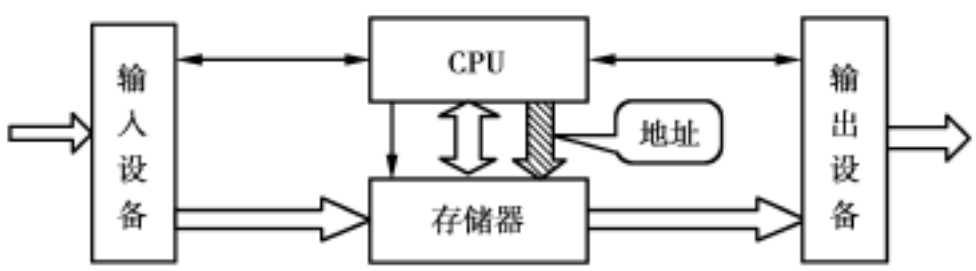


图 1.2　计算机硬件的组成

图 1.2 是计算机硬件的组成示意图，粗箭头是数据流向，细箭头是控制信号流向。输入、输出设备将自己的状态送给 CPU 中的控制器，控制器发出控制信号，控制输入、输出设备和存储器。在控制器的控制下，数据或程序由输入设备输入，并存放到存储器。CPU 发出地址信号，取出并执行在存储器中的指令，从存储器中取出数据进行处理，处理结果存放到存储器。存储器中的数据也可以在 CPU 的控制下传送到输出设备，由输出设备输出。

1.2.2　程序的工作原理及其在计算机中的运行过程

在上述计算机的硬件组成中 CPU 和存储器是关键的部件，计算机的程序以机器语言的形式存放在存储器中，程序要处理的数据也存放在存储器中。计算机的内存由许多具有连续编号的存储单元所组成，单元以“字节”为单位，每一个单元的编号都是惟一的。如同一条街道一样，每个门都有一个惟一的门牌，门牌的编号是按顺序连续编写的。这个编号在计算机中就是“地址”。可以这样理解程序在计算机中的存储（参见图 1.3）：程序通过输入设备输入计算机，被翻译成机器语言后，按照程序语句的顺序存放在存储器的一个区域内，而程序处理的数据存放在另一个存储区域中；在 CPU 中有一个专门的计数器（叫程序计数器 PC 或指令指针 IP）存放下一条要执行的语句的地址，CPU 每执行一句，这个计数器就会自动装入下一条要执行的语句的地址。

图 1.3 中程序的执行过程可以理解为如下步骤：①由 CPU 中的程序计数器 PC 向存储器

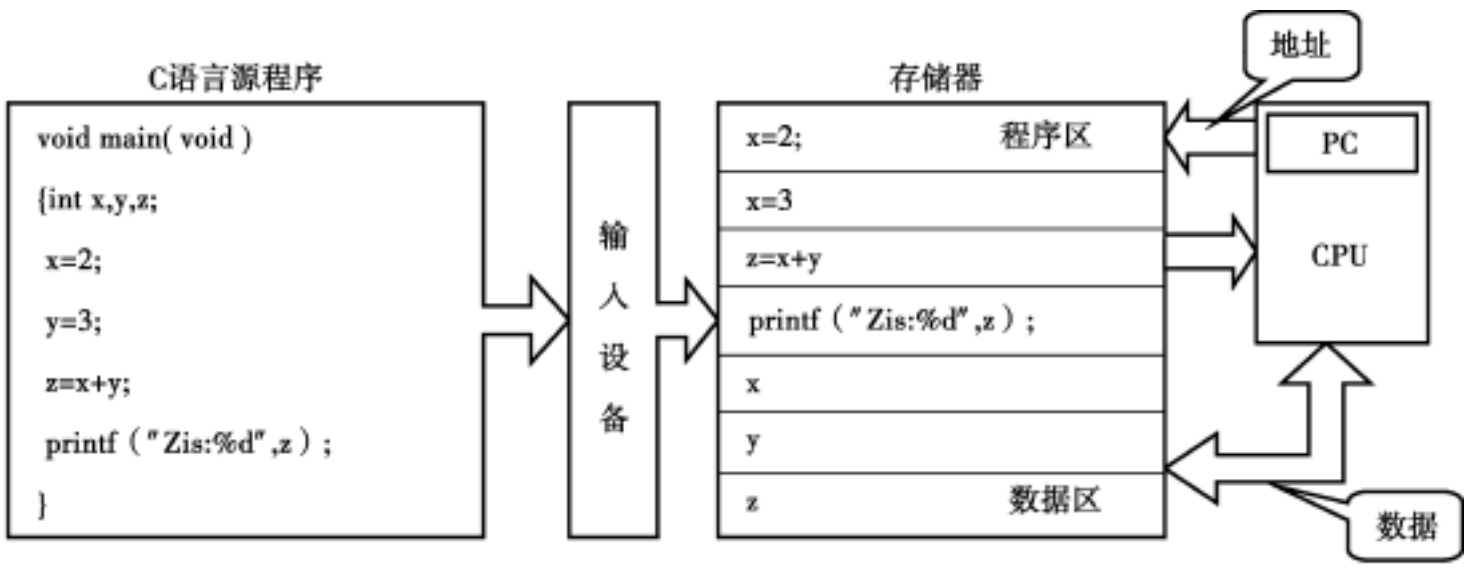


图 1.3　程序在计算机中的存储

发出地址信号，CPU 将第一句程序“ x = 2; ”读入；②CPU 对这条语句进行分析，解释这条语句的意义；③CPU 执行这条语句，将数据区中存放“ x ”这个变量的存储单元中的数据改写成 2，并且将第二句“ y = 3; ”的地址存到 CPU 中 PC 里去；④程序计数器 PC 向存储器发出地址信号，CPU 将第二句程序“ y = 3; ”读入，重复上述步骤。计算机就是这样不断地重复取指令、分析指令、执行指令这 3 步，程序就逐条语句地执行下去。

1.3　C 语言的特点

C 语言是美国 Bell 实验室开发成功的。它在实践中表现出了强大的生命力，在当时高级语言基本上都不适合开发系统软件，而 C 语言却成功地开发出了 Unix 操作系统，而且绝大多数由 Unix 提供的软件工具，包括 C 编译器本身都是用 C 语言编写的。它的表达式简洁，具有丰富的运算符和良好的控制结构与数据结构，成为结构化程序设计的主流语言。目前 C 簇的程序设计语言 C 和 C++ 是当前最流行的程序设计语言，拥有广泛的用户群，甚至被认为是计算机专业程序员必须掌握的语言。学习 C 语言也可以作为学习 C++ 和 JAVA 的基础。

C 语言之所以能存在和发展并具有强大的生命力，关键在于它有许多不同于（或优于）其他语言的特点。C 语言的主要特点如下:

1）语言简洁、紧凑，使用方便、灵活。C 语言一共只有 32 个关键字，程序书写形式自由。C 语言具有结构化的 9 种控制语句，用函数作为程序的模块单位，便于实现程序的模块化。

2）数据结构丰富。C 语言具有整型、实型、字符型、数组类型、指针以及结构、联合、枚举等自定义类型等，能用来实现各种复杂的数据结构（如链表、树、栈等）的运算。尤其是指针的功能和作用非常强大。C 语言是高级语言，却能支持位和字节直接操作，可以直接访问硬件，这点是其他高级语言所不具备的，是它的“ 中级语言 ”特性，也是它强大能力的基础。

3）运算符丰富。C 的运算符包含的范围很广泛，共有 34 种。C 把括号、赋值、强制类型转换等都作为运算符处理，从而使 C 的运算类型极其丰富，表达式类型多样化。灵活使用各种运算符可以实现在其他高级语言中难以实现的运算。

4）生成的目标代码紧凑、高效。C 语言的代码效率一般要高于其他高级语言编译产生的代码，而且 C 语言的一些低级特性为编程者改善特定程序段的效率提供了可能，这对编写实时系统有相当重要的意义。

5）良好的标准化程度和可移植性。在各种不同操作系统和硬件平台上的 C 语言之间保持了相当兼容的水平。

6）容易入门，全面掌握较难。C 语言和其他高级语言一样，可以很容易入门，初学者很快就可以编写出一些小程序。但是，由于 C 语言编程十分灵活、语法限制少、功能强大，与其他高级语言相比，要全面掌握 C 语言则比较难。

由于 C 的以上特点，C 和 C 族编程语言被广泛应用于各种系统软件、应用软件和组件的开发。由于 C 语言的广泛应用，新的 C 和 C 族语言编译系统不断出现，同时也使它成了首选的教学语言之一，被广泛使用于程序设计教学中。总之，C 语言是面向程序员的语言，对程序员要求较高，程序员使用 C 语言编写程序会感到限制少、灵活性大、功能强，可以编写出任何类型的程序。现在，学习、使用 C 和 C 族语言的人已越来越多。

1.4　简单 C 程序

下面是两个简单的 C 语言程序，可以从中体会一下 C 语言的程序是怎样构成的。

例 1.1

```
#include〈stdio. h〉
void main (void)
{
    printf ("Hello!");
}
```

或写成:

```
main ( )
{
    printf ("Hello!");
}
```

例 1.1　是一个最简单的 C 语言程序，它在屏幕上显示为“Hello!”。

第一个程序是比较严格的写法，程序的第一行表示：在程序前包含 stdio. h 文件，在这个文件中定义了许多实现输入/输出功能的“函数”，程序中用到的 printf 就是在 stdio. h 文件中声明的。程序的第二行声明了 main 函数，这是程序的“主函数”。每个程序都有一个而且只有一个主函数，主函数是程序开始执行的入口，主函数名必须为 main。main 前的“void”(空) 表示这个主函数不返回数值，小括号中的“void”表示这个函数没有参数。程序第三行和第五行的“{”与“}”是一对，在 C 程序中，须用“{}”将函数括起来。程序的第四行调用库函数 printf 在屏幕上显示“Hello!”，该行向右边缩进两个字符是为了便于阅读。

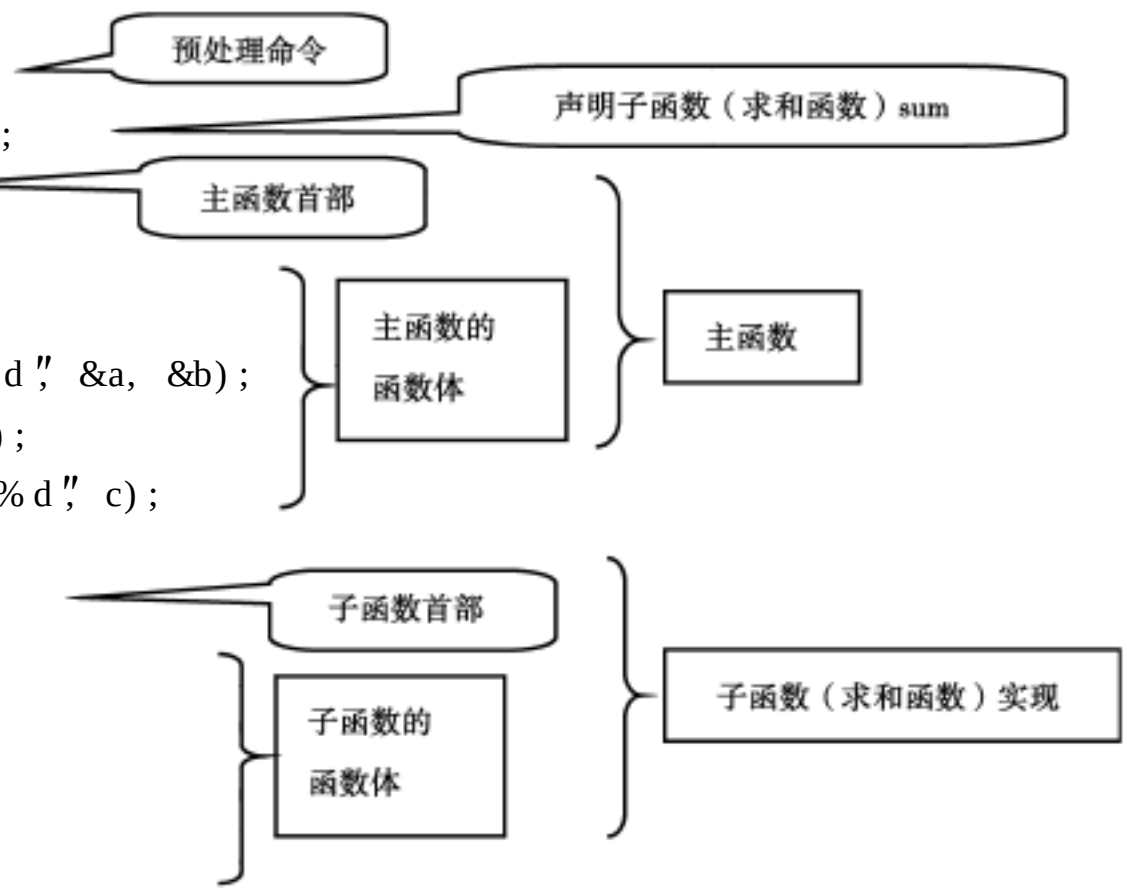
对于一些编译系统如 Turbo C 可以写成第二个程序那样，与第一个程序相比，省掉了第

一行，省掉了 main 前和小括号中的“ void ”。严格地说，这两个程序不太一样，省掉了 main 前“ void ”表示 main 函数将返回一个整形数，省掉第一行在某些编译器中会出现错误或警告，如第二个程序在 Visual C++ 中会出现一个警告，提示 printf 没有定义；而小括号中的“ void ”总是可以省掉的。笔者建议不要省掉第一行和函数名前的“ void ”，从初学开始就培养严谨而良好的程序书写风格。

例 1.2 是一个比较完整的程序，它完成从键盘上输入两个整数，计算并显示它们的和。

例 1.2

```
#include〈stdio. h〉
int sum  ( int x,  int y) ;
void main  ( void)
{
        int a,  b,  c;
        scanf  ( "% d, % d" ,  &a,  &b) ;
        c = sum  ( a,  b) ;
        printf  ( "sum = % d" ,  c) ;
}
int sum  ( int x,  int y)
{
        int z;
        z = x + y;
        return  ( z) ;
}
```



在例 1.2 中，主函数定义了 3 个整型“ 变量 ”a，b，c，调用库函数 scanf 从键盘上输入两个整数分别存放在变量 a 和 b 中，调用子函数 sum 完成 a 与 b 相加的计算，并将结果存放在变量 c 中，最后调用库函数 printf 将结果输出。子函数 sum 有两个参数 x 和 y 分别接收主函数中 a 和 b 的值。定义了一个变量 z，用于存放 x 与 y 相加的和，并用 return 返回给主函数。程序运行情况如下:

在键盘上键入:　　23，9 ↙

程序在屏幕上显示:　　sum = 32

上面的“ ↙ ”表示回车键。

从例 1.1 和例 1.2 可以看出，C 语言程序由许多函数构成，包含一个主函数和任意多个子函数。函数是 C 程序的基本单位，故 C 语言又被称为函数式语言。被调用的函数可以是用户自定义函数，如例 1.2 中的 sum，也可以是库函数，如 printf 和 scanf。库函数是编译系统自带的，除了 ANSI C 提供的 100 多个库函数外，不同的编译系统均增加了许多不同的库函数。

从程序的结构上看，C 语言程序可以分为如下几个部分:

①以“ # ”号开头的预处理命令 (有关预处理命令将在第 7 章中学习) 。

②全局声明 (在例 1.2 中声明了一个自定义子函数 sum) 。

③函数名为 main 的主函数，程序总是从 main 函数开始执行。

④其他自定义函数。自定义函数和主函数的放置位置没有先后次序，主函数可以放在自定义函数的前面或后面，也可以在主函数的前面或后面放置一些自定义函数。

除了 main 函数外，每个自定义函数都需要在全局声明处给予声明。C 语言的变量、函数均是先声明后使用，但也有的编译系统不太规范，如 Turbo C 的自定义函数就可以不用事先声明。

每个函数的实现部分都可以分为两个部分：函数的首部和函数体。

函数首部由函数的返回类型、函数名、小括号对、参数类型、参数名组成，例如例 1.2 的 sum 函数首部如下：

```
int      sum     (  int     x ,     int     y )
 ↓        ↓        ↓        ↓        ↓        ↓
返回值类型  函数名  参数类型  参数名  参数类型  参数名
```

函数的返回类型可以缺省，返回类型缺省时，函数的返回类型是整形。注意：函数首部与函数声明在格式上几乎相同，只是函数声明时在最后加了一个“；”号。

函数体用一对花括号“｛｝”括起来。在花括号对中，函数体可以分为定义部分和执行部分，定义部分用来声明函数中要用到的变量，如例 1.2 中的语句“int z;”和“int a, b, c;”。执行部分是可执行的语句，在 C 语言中每个 C 语句均以“；”号结束。

1.5　C 语言的编译连接

在 1.1 节中已经学过：任何语言编写的程序都要翻译成二进制机器语言的程序，计算机只认识机器语言。在计算机中，C 语言的源程序是以文件形式存放在磁盘上的，通常有*.h 和*.c 两种扩展名，扩展名为.h 的是首标文件，也叫头文件，扩展名为.c 的是源文件，这两种文件都是文本文件，可以用各种文本编辑器编辑。这些源程序如何翻译成机器语言的程序呢？一般要经过如下几个步骤：

①利用文本编辑软件录入和编辑源程序，产生扩展名为.c 或.h 的文件，以例 1.2 为例，可以给它取名为：example2.c。

②接着用编译程序编译 example2.c，产生 example2.obj 文件，这个文件称为目标文件或叫目标块，它已经是机器语言的程序了，但还不能直接运行。

③再用连接程序将 example2.obj 与所用到的库函数等模块连接起来，并生成可执行文件 example2.exe。这个文件在操作系统的支持和管理下运行，例如在 WINDOWS 下可以用命令行方式或鼠标点击后装入内存运行。

对于不同的机器和操作系统，这个过程也会有一些差异。如果是 WINDOWS 程序，还需要用资源编辑器输入和编辑位图、菜单、图标、快捷键等资源，这些资源经过编译后绑定到可执行文件上。如果还使用了 ActiveX 控件的话，控件还需要注册。现在的编译系统基本上都将程序的编辑软件、编译软件、连接软件和调试软件集成在一起，形成一个有机的整体，做成集成开发环境。程序员可以方便地在窗口方式下连续进行编辑、编译、连接、调试、运行的全过程，例如常用的 Visual C++，Turbo C 等都有很方便的集成环境（Visual C++ 集成环境上机操作参见附录）。

1.6 C 语言学习建议

C 语言程序设计是一门实践性很强的课程，除了像学习别的课程那样要预习和复习、做好每一次作业外，学习 C 语言要非常重视上机实验，不上机做实验是学不好 C 语言的。要认真做好每一次实验前的准备工作，预先编写好要上机的程序，这样上机时才能有充裕的时间去调试和改正错误，才能正确地完成实验。下面是给初学者的一些建议：

1）养成良好的程序书写风格。包括：结构层次缩进风格、符号书写风格、必要的注释等。好的书写风格可以使程序易于阅读、扩充、修改和维护，最终使程序更加稳定。

2）注意写好程序的每一句。大程序是由一行行代码构成，在写每一句代码时想一想：这句为什么要这样写？为什么要放在这里？可以不要吗？可以把自己当成 CPU，在大脑中一步步运行程序。

3）经常读别人编的好程序。好的程序就是一件艺术品，读之能体会别人设计程序的用心和思想，久而久之，自己也能编出这些程序。

4）从简单到复杂，模仿书上或别人编好的程序。任何人学习程序设计都是从模仿开始，直到编写自己的作品。模仿的目的是为创造自己的作品打基础。

5）熟练记住几个算法。就像学习英语不仅要背单词而且还要背句型一样，这几个算法是不得不记住的：查找（顺序查找、折半查找）、排序（选择排序、冒泡排序）、字符串拼接与复制等。

6）努力学习并精通程序调试和排错的能力。包括：程序的编译、连接、断点设置、单步跟踪、变量的查看等。初学者最困惑的是程序出了错而不知道错在哪儿，可以试着把对的程序故意改错，再观察出错信息，这样可以了解错误信息产生的真正原因。

7）做一个较大作业（至少 1 000 行代码左右）。在这个作业中，能学会几件很重要的东西：模块化程序设计的好处，模块与模块之间信息的传递方法，怎样使程序至顶向下、从无到有直到变得很庞大（建议选择本书第 13 章中的例子为题，完成整个程序）。

习　题

1. 叙述计算机程序的工作原理。

2. 设想有如下问题：从键盘上输入一个圆的半径，计算该圆的周长和面积，在屏幕上显示结果。试描述计算机执行该程序的步骤和过程。

3. 简述 C 语言程序的编写步骤。

4. 简述 C 语言程序的结构。

5. C 语言的书写有何特点？

6. 使用并熟悉一种 C 语言编译集成环境，并编写和运行本章的例 1.1 和例 1.2。

第 2 章
数据类型、表达式

万丈高楼平地起,一个无论多么复杂的软件,都是由一行行代码组成的,每一行代码又是由一些最基本的元素按照一定的规律组成的。要学习编写程序,就要从这些最基本的元素和组成规律学起,这些最基本的元素是:标识符、变量、常量、运算符、关键字、表达式等。

2.1 C 的数据类型

从“计算机”这个名称就可以想到:计算机可以计算、处理和加工信息。因此,一个程序应包括两个方面的内容:

①对数据的描述,即对数据和数据的组织形式(也称为“数据结构”)进行定义和描述。

②对操作步骤的描述,即对数据的处理过程,也称为“算法”。

计算机科学家 Nikilaus Wirth 提出一个公式:

程序 = 数据结构 + 算法

在这个公式中,算法是程序的灵魂,数据结构是加工的对象。在 C 语言中对数据结构基本的描述就是数据类型的定义。在本章中学习对数据的最基本描述,主要介绍基本数据类型。C 的数据类型如图 2.1 所示。

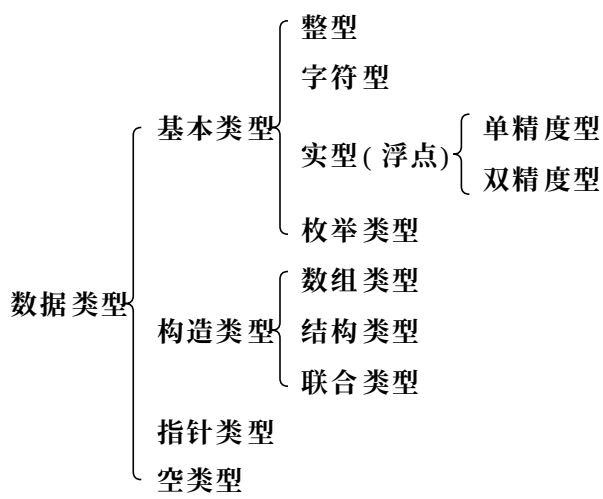


图 2.1 C 的数据类型

图 2.1 中基本类型是其他类型的基础,不同的类型在计算机存储器中占用的存储空间不同,表示数据的值的范围也不同,这两者是相互关联的。构造类型是在基本类型的基础上由程