主编　密君英

C语言程序设计基础

苏州大学出版社

C语言程序设计基础

主　编　密君英
副主编　刘海明　　方　蓓

苏州大学出版社

图书在版编目(CIP)数据

C语言程序设计基础/密君英主编. —苏州：苏州大学出版社，2005.12
ISBN 7-81090-602-X

Ⅰ.C… Ⅱ.密… Ⅲ.C语言-程序设计-高等学校；技术学校-教材 Ⅳ.TP312

中国版本图书馆CIP数据核字(2006)第003024号

C语言程序设计基础

密君英 主编

责任编辑 周建兰

苏州大学出版社出版发行
（地址：苏州市干将东路200号 邮编：215021）
丹阳兴华印刷厂印装
（地址：丹阳市胡桥镇 邮编：212313）

开本 787mm×1 092mm 1/16 印张 14.5 字数 361千
2005年12月第1版 2005年12月第1次印刷
ISBN 7-81090-602-X/TP·42(课) 定价：33.00元

苏州大学版图书若有印装错误，本社负责调换
苏州大学出版社营销部 电话：0512-67258835

前　言

C 语言因其功能强大、内容丰富、使用灵活、程序执行效率高、可移植性好、既具有高级语言的特点、又具有汇编语言的一些特点，长期以来一直备受广大程序设计爱好者的青睐。在面向对象的程序设计理念流行的今天，有些人认为结构化程序设计已没有必要再专门学习，但作为程序设计的入门课程，我们必须严格地培养学生的编程习惯和思维，如一开始就让学生接触 Java、C ++ 等语言则有点困难。无疑 C 语言成了一门最好的入门语言，严谨而又灵活的语法规则，必定能养成学生良好的编程思路和编程习惯，为学习其他高级语言奠定良好的基础。

全书共分 12 章，主要包括 C 语言概述、数据类型、运算符和表达式、顺序结构、选择结构、循环结构、数组、函数、指针、编译预处理、结构体与共用体、位运算、文件等。本书在表达上尽量采用通俗易懂的语言，以确切、详尽的例题来说明问题，使读者形象、快捷地掌握知识点并能够熟练运用，并在每章的最后配套了大量的、形式多样的习题，能帮助学生更好地学习和巩固所学知识。

本教材编写时注意以实例为主，强调了理论与实际结合，特别适合高职高专的学生学习。当然，本书也可作为其他学生学习 C 语言的参考书和等级考试的参考教材。

本书由密君英主编，刘海明、方蓓为副主编，吴凯益为参编，其中第一、二、三章由密君英编写，第四、五章由吴凯益编写，第六、七、十一章由方蓓编写，第八、九、十、十二章由刘海明编写。徐晓明教授在百忙之中通阅了全书，并提出了许多宝贵的建议，在此深表感谢。在本书的编写过程中，还得到了苏州大学李凡长教授、苏州农业职业技术学院电子系相关老师的指导与帮助，在此一并表示感谢。

由于作者水平有限，加上时间仓促，书中的错误和不足在所难免，我们真诚希望得到广大读者的批评指正。

编　者

2005 年 10 月

目　录

第 1 章

C 语言概述

本章要点：

- 了解 C 语言的发展及特点。
- 了解几种典型的 C 程序结构。
- 掌握结构化程序的特点。
- 了解 C 程序的上机步骤。

1.1　C 语言的发展及特点

1.1.1　C 语言的发展简史

C 语言是国际上广泛流行的计算机高级语言，既可以用来编写系统软件，也可以用来编写应用软件。

在 C 语言诞生以前，系统软件主要是用汇编语言编写的。由于汇编语言程序依赖于计算机硬件，其可读性和可移植性都很差；为了提高程序的可读性和可移植性，最好改用高级语言，但一般的高级语言又难以实现对计算机硬件的直接操作（这正是汇编语言的优势），因此，人们盼望有一种兼有汇编语言和高级语言特性的新语言。于是，C 语言就在这种情况下应运而生了。

C 语言是在 B 语言的基础上发展起来的。1970 年，美国贝尔实验室的 Ken Thompson 设计出了很简单的而且很接近硬件的 B 语言，并用 B 语言编写了第一个 UNIX 操作系统。但由于 B 语言过于简单，功能有限，1972 ~ 1973 年间，贝尔实验室的 D. M. Ritchie 在 B 语言的基础上设计出了 C 语言。C 语言既保持了 B 语言精炼、接近硬件的优点，又克服了它们过于简单、数据无类型的缺点。

最初的 C 语言只是为描述和实现 UNIX 操作系统提供一种工作语言而设计的，后来 C 语言又作了多次改进，但主要还是在贝尔实验室内部使用。直到 1975 年，UNIX 第 6 版公布后，C 语言突出的优点才引起人们的普遍注意。于是，1977 年出现了不依赖于具体机器的 C 语言编译文本《可移植 C 语言编译程序》，使 C 语言移植到其他机器时所需做的工作大大简化了，这同时也推动了 UNIX 操作系统在各种机器上的实现。应该说，C 语言和 UNIX 操作系统是紧密相连的，在发展过程中相辅相成。1978 年，C 语言先后移植到大、中、小微机上时，已独立于 UNIX 操作系统了。现在，C 语言在全世界的应用非常普及，成为世界上应用

最广泛的几种计算机语言之一。

以 1978 年发表的 UNIX 操作系统第 7 版中的 C 语言编译程序为基础，Brian W. Kernighan 和 Dennis M. Ritchie（合称 K&R）合著了影响深远的名著《The C Programming Language》。这本书中介绍的 C 语言成为后来广泛使用的 C 语言版本的基础，它被称为标准 C。1983 年，美国国家标准化协会（ANSI）根据 C 语言问世以来各种版本对 C 的发展和扩充，制定了新的标准，称为 ANSI C。ANSI C 比原来的标准 C 有了很大的发展，K&R 在 1988 年修改了他们的经典著作《The C Programming Language》，按照 ANSI C 标准重新编写了该书。1987 年，ANSI 又公布了新标准——87 ANSI C。1990 年，国际标准化组织 ISO 接受 87 ANSI C 为 ISO C 的标准，目前流行的 C 语言编译系统都是以它为基础的。本书基本上以 ANSI C 为基础。目前，广泛流行的各种版本的 C 语言编译系统基本部分虽然是相同的，但也有一些不同。在微机上使用的有 Microsoft C、Turbo C、Quick C、Borland C 等，它们的不同版本又略有差异，因此，读者在使用前应先了解所用的计算机系统配置的 C 语言编译系统的特点和规定，可以查阅相应的手册。

1.1.2 C 语言的特点

C 语言同时具有汇编语言和高级语言的优势。概括起来，C 语言主要有以下特点：

● 语言简洁、紧凑，使用方便、灵活。C 语言共有 32 个关键字（见附录）、9 种控制语句，程序书写形式自由，主要用小写字母表示，压缩了一些不必要的成分。表 1.1 列举了 C 语言与 PASCAL 语言之间的主要区别，请读者注意比较（表 1.1）。

表 1.1

C 语言	PASCAL 语言	含　义
{ }	BEGIN…END	复合语句
if (e) S	IF(e) THEN S	条件语句
int i;	VAR i :INTEGER	定义 i 为整型变量
int a[10]	VAR a:ARRAY[1..10] OF INTEGER	定义 a 为整型一维数组
int f( )	FUNCTION f ( ):INTEGER	定义 f 为返回整型值的函数
int * p	VAR p: ↑ INTEGER	定义 p 为指向整型变量的指针变量
i+=2	i: = i+2	赋值语句，使 i+2→i
i++, ++i	i: = i+1	自增值语句，使 i+1→i

● 运算符极其丰富。C 语言的运算符很多，共有 34 种运算符（见附录）。C 语言把括号、赋值、强制类型转换等都作为运算符处理，运算类型极其丰富，表达式类型多样化，从而可以在 C 语言中实现在其他高级语言中难以实现的运算。

● 数据结构丰富，具有现代化语言的各种数据结构。C 语言的数据类型有整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型等，能用来实现各种复杂的数据结构。

● C 语言是良好的结构化设计语言，具有结构化的控制语句（如 if…else 语句、while 语句、do…while语句、switch 语句），用函数作为程序的模块单位，便于实现程序的模块化。

● 语法限制不太严格，程序设计自由度大。如对数组下标越界不作检查，由程序编写者自己掌握程序的正确性。整型数据和字符型数据之间可以通用。C 语言由于允许程序编写者有较大的自由度，因而放宽了对语法的检查。因此，程序员要仔细检查程序，保证其正

确,不要过分依赖于 C 语言编译程序去查错。一般来说,使用 C 语言比使用其他高级语言对程序设计的熟练度有更高要求。

● 生成的目标代码质量高,程序执行效率高。C 程序一般只比汇编程序生成的目标代码效率低 10% ~20%。

● 与汇编语言相比,程序可移植性好。C 程序基本上可以不做修改就能用于各种型号的计算机和各种操作系统。

● C 语言能直接操纵硬件。C 语言能进行位(bit)操作,能实现汇编语言的大部分功能,能直接对硬件进行操作。C 语言既具有高级语言的功能,又具有低级语言的许多功能,因此可以用来编写系统软件。有人把 C 语言称为"高级语言中的低级语言"或"中级语言",一般仍认为其是高级语言,因为 C 程序也要经过编译、连接才能得到可执行的目标程序。

上面这些是 C 语言的一般特点,至于 C 语言内部的一些特点,我们将在后面的章节中结合具体内容进一步介绍。

总之,C 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,灵活度大,功能强,可以编写出任何类型的程序。现在 C 语言不仅可以用来编写系统软件,也可以用来编写应用软件。C 语言因其具有理想的结构化设计语言,现在普遍用做教学语言,受到大多数人的青睐。

1.2　结构化程序设计的基本概念

一般我们在进行程序设计时,关键的一步是设计算法(即为解决一个问题而采取的确定且有限的步骤)。有了一个好的算法,就可以用任何一种计算机高级语言将算法转换为程序。算法一般用伪代码和流程图来表示。常用的流程图由以下几种基本框组成,如图 1.1 所示。

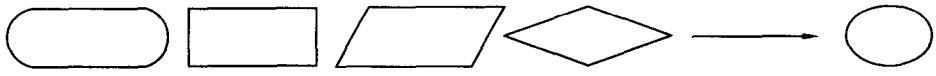

图 1.1　基本流程框图

用这些框和流程线组成的流程图来表示算法,形象直观,简单方便。但是这种流程图对于流程线的走向没有任何限制,可以任意转向,在描述复杂的算法时所占篇幅较多,费时费力且不易阅读。随着结构化程序设计的出现,1973 年,美国学者 L. Nassi 和 B. Shneiderman 提出了一种新的流程图形式。这种流程图完全去掉了流程线,算法的每一步都用一个矩形框来描述,把一个个矩形框按执行的次序连接起来就是一个完整的算法描述,这种流程图用两位学者名字的第一个英文字母命名,称为 N-S 流程图。

下面简单介绍一下结构化程序的三种基本结构,分别用传统流程图和 N-S 流程图表示。

1.2.1　顺序结构

顺序结构是最简单的一种结构,程序中的语句是按其在程序中的先后顺序逐条执行的,没有分支和转向。一般赋值语句、输入/输出语句可以构成顺序结构。图 1.2 是顺序结构流

程图,(a)为一般流程图,(b)为 N-S 流程图。

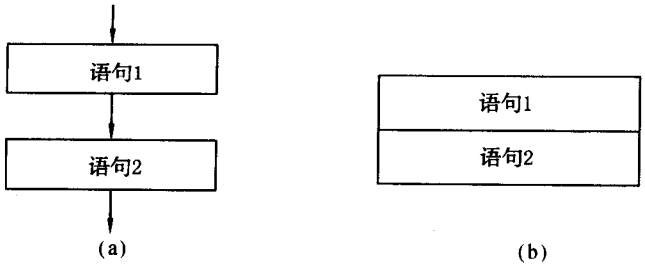

图 1.2 顺序结构

1.2.2 选择结构

选择结构或称分支结构,一般 if 语句、switch 语句可构成选择结构。程序中执行到这些语句时,将根据不同的条件去执行不同分支中的语句。图 1.3 是选择结构流程图,(a)为一般流程图,(b)为 N-S 流程图。

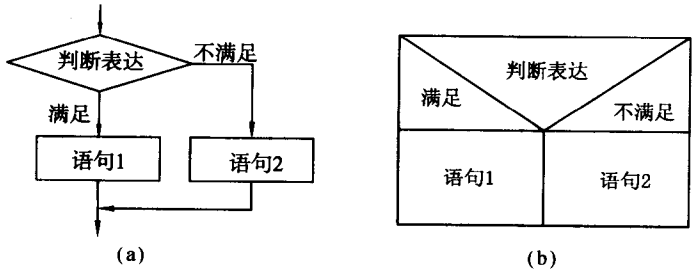

图 1.3 选择结构

1.2.3 循环结构

循环结构又称重复结构,即反复执行某一操作。循环结构有两种类型:当型循环和直到型循环。当型循环的特点是:当指定的条件满足(成立)时,就执行循环体,否则不执行;直到型循环的特点是:执行循环体,直到指定的条件满足(成立)时,就退出循环。图 1.4 为当型循环流程图,图 1.5 为直到型循环流程图。其中,(a)为一般流程图,(b)为 N-S 流程图。

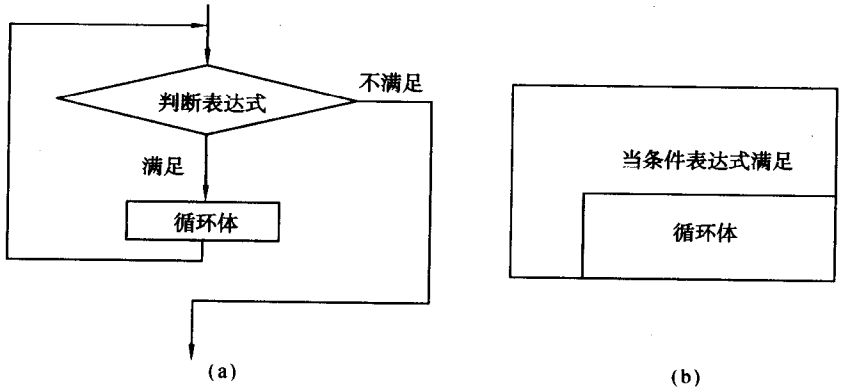

图 1.4 当型循环

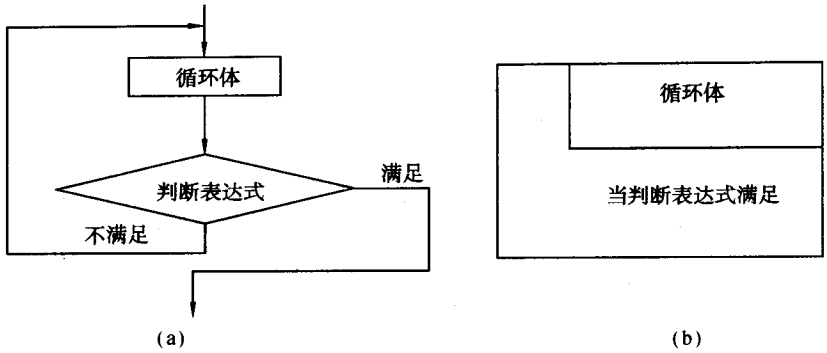

图 1.5　直到型循环

以上是结构化程序设计的三种基本结构，C 语言是一种结构化程序设计语言，它直接提供了这三种基本结构的语句。在后续章节中我们将分别介绍这些内容。

1.3　简单的 C 程序介绍

下面先通过几个简单的 C 程序，介绍 C 程序的一些基本构成和格式，使读者对 C 语言程序有一个初步的了解。

例 1.1

```
main( )
{
  printf("This is a C program. \n");
}
```

本程序的运行结果如下：

```
This is a C program.
```

其中，main 是主函数名，C 语言规定每个 C 程序必须有一个主函数，且只能用 main 表示主函数名，其后的一对括号表示函数参数，中间可以是空的，但括号不能省略。main( ) 是主函数的起始行，一个 C 程序可以包含多个不同名的函数，但主函数只能有一个，一个 C 程序总是从主函数开始执行。

在函数的起始行下面是函数体，函数体用一对{ }括起来，本例中函数体只有一个输出语句，有关 printf 的具体介绍详见第 3 章。注意，在 printf 语句最后有一个分号。C 语言规定程序中的每一条语句都必须用分号“;”结束，分号是 C 语句的一部分，不是语句之间的间隔符。

例 1.2

```
main( )                                  /* 求矩形面积 */
{
  float a,b,area;                        /* 定义变量 */
  a=1.2;b=3.6;                           /* 给矩形的两条边赋值 */
  area=a*b;                              /* 求出矩形的面积,并存入变量 area 中 */
  printf("a=%f,b=%f,area=%f\n",a,b,area); /* 输出矩形的两条边长和面积 */
}
```

本程序的运行结果如下：

a = 1.200000, b = 3.600000, area = 4.320000

本程序的作用是求出矩形面积，/ * …… * /表示注释部分，可用中文或英文作注释。注释只是为了便于别人能更好地理解程序，对编译和运行不起作用。注释可以加在程序中的任意位置。函数体中第 1 行是声明部分，定义变量 a、b、area 为实型变量；第 2 行是两个赋值语句，使 a 和 b 的值分别为 1.2 和 3.6，C 程序中多条语句可写在同一行，每条语句后必须加上分号；第 3 行是计算 a 和 b 的积并赋给变量 area；第 4 行是输出语句，其中%f 是输入/输出的格式字符串，有关这部分介绍详见第 3 章。

例 1.3

```
int max(int x,int y)
  /*定义 max 函数,函数值为整型,形式参数 x、y 为整型*/
{
  return(x>y? x:y);
  /*返回 x 和 y 中较大的数,并通过 max 函数带回调用处*/
}

main()                                  /*主函数*/
{
  int num1,num2;          /*声明部分,定义变量 num1 和 num2 为整型变量*/
  printf("Input the first integer number:");
  scanf("%d", &num1);                   /*输入变量 num1 的值*/
  printf("Input the second integer number:");
  scanf("%d", &num2);                   /*输入变量 num2 的值*/
  printf("max=%d\n", max(num1, num2));
  /*输出 num1 和 num2 中的较大数*/
}
```

本程序的运行结果如下：

```
Input the first integer number: 6
Input the second integer number: 9
max=9
```

在这个程序中，有两个函数：main 函数和被调用的函数 max。max 函数的作用是将 x 和 y 中较大的数，通过 return 语句将值返回给主调函数 main，返回值是通过函数名 max 带回到函数的调用处的。主函数中 scanf( ) 是输入函数，其作用是输入变量 num1 和 num2 的值，有关 scanf( ) 的介绍详见第 3 章。本例中用到了函数调用、实际参数和形式参数等概念，读者在此可不必予以深究，在学到后续章节时，问题自然就迎刃而解。

通过上述几个例子，可以得出如下结论：

● C 程序是由函数构成的，函数是 C 语言程序的基本单位。一个 C 源程序至少包含一个 main 函数，也可以包含一个 main 函数和若干个其他函数。main 函数的作用相当于其他高级语言中的主程序，其他函数的作用相当于子程序。编写 C 程序就是编写一个个函数。

● 一个函数一般由两部分组成。

◇ 函数的首部，即函数的第一行。包括函数名、函数类型、函数参数名、参数类型。例如，例 1.3 中的 max 函数的首部为

```
函数类型      函数名      函数参数类型        函数参数名
   ↓           ↓            ↓                  ↓
  int        max (        int   x,           int    y   )
```

一个函数名后面必须有一对括号“( )”，参数可以没有，如 main( )。

◇ 函数体，即函数首部下面{……}内的部分。如果一个函数有多对{}，则最外层的一对{}为函数体的范围。函数体内一般有声明部分和执行部分。声明部分主要是对函数中的变量进行定义，函数体中的变量定义语句必须在所有可执行语句之前。执行部分由若干个语句组成，在某些特殊情况下，函数体内可以没有声明部分，也可以声明部分和执行部分都没有。例如：

```
dump( )
{  }
```

这是一个空函数，不执行任何操作，但它是一个合法函数。

● 一个 C 程序总是从 main 函数开始执行，不论 main 函数在整个程序中的位置如何。可以把 main 函数放在程序最前面，也可以放在程序最后，或者放在某些程序之前、某些程序之后。当主函数执行完毕，也即程序执行完毕。习惯上，把 main 函数放在程序最前面。

● C 程序书写格式比较自由，可以一行上写多个语句，也可以把一个语句写在多行上。C 程序语句没有行号，对于从哪一列开始书写也无特殊规定。

● 每条语句必须以分号结束，函数的最后一条语句也不例外，分号是 C 语句的组成部分。

● C 语言本身没有输入/输出语句，输入和输出的操作是通过库函数 printf 和 scanf 等函数来完成的。

● 程序中可以用/*……*/对 C 程序语句进行注释，以增加程序的可读性。“/*”和“*/”必须成对使用，且“/”和“*”以及“*”和“/”之间不能有空格，否则会出错。注释的位置可以独占一行，也可以跟在语句的后面，如果一行写不下，可另起一行继续写。

技巧：为避免可能会遗漏必须配对使用的符号，如注释符号、函数体的起止标识符{}和( )等，在输入时可连续输入这些起止标识符，然后再在其中进行插入来完成内容的编辑。在起止标识符嵌套以及它们相距较远时，这样做很有必要。

1.4　C 程序的上机步骤

前面我们已经介绍了一些用 C 语言编写的程序。要使计算机能按照人的意志进行工作，必须根据问题的要求，编写出相应的程序。所谓程序，就是一组计算机能识别和执行的指令。每一条指令使计算机执行特定的操作，程序可以用高级语言（如 QBASIC、FORTRAN、PASCAL、C 等）编写。用高级语言编写的程序称为源程序（Source Program）。从根本上说，计算机只能识别和执行由“0”和“1”构成的二进制指令，而不能识别由高级语言编写的指令。为了使计算机能识别和执行高级语言源程序，必须用一种称为“编译程序”的软件，将源程序编译成二进制形式的“目标程序”，然后将该目标程序和系统的函数库以及其他目标