



工业和信息产业科技与教育专著出版资金资助出版
工业和信息化部高等教育“十二五”规划教材

C语言 程序设计

● 主 编 姚立新 梁宏涛



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

工业和信息产业科技与教育专著出版资金资助出版

工业和信息化部高等教育“十二五”规划教材



语言 程序设计

主 编 姚立新 梁宏涛

副主编 苏爱玲 林旭平 姜 琴

杨新艳 房正华

电子工业出版社

Publishing House of Electronics Industry

北京 • BEIJING

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，侵权必究。

图书在版编目(CIP)数据

C 语言程序设计 / 姚立新, 梁宏涛主编. —北京: 电子工业出版社, 2013.8

ISBN 978-7-121-21102-7

I. ①C… II. ①姚… ②梁… III. ①C 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2013)第 174548 号

策划编辑: 郝国栋

责任编辑: 郝国栋

印 刷: 三河市鑫金马印装有限公司

装 订: 三河市鑫金马印装有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱

邮编: 100036

开 本: 787×1092 1/16 印张: 15.5 字数: 360 千字

印 次: 2013 年 8 月第 1 次印刷

定 价: 29.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

前 言

C 语言是一种面向过程的计算机程序设计语言,是目前流行的结构化程序设计语言之一。

C 语言简洁、紧凑、灵活,数据类型丰富,功能强大,可移植性好。目前几乎所有高等学校理工科专业都把 C 语言作为第一门编程语言课程,本书作为适用本科低年级学生初学程序设计及 C 语言的教材,以掌握 C 语言基本语法、培养程序设计思维、提高上机实践能力为目标,旨在以简单明了的语言和简洁实用的例子,阐述 C 语言的基本语法和程序设计的基本逻辑思路。C 语言的语法及使用有很多灵活和复杂的方面,对初学者来说往往难以全面掌握,所以我们只对主要的适合初学者入门的知识点进行讲解,主要目的是通过 C 语言的学习与实践,培养学生程序设计的抽象逻辑思维能力,掌握动手编程上机实践的技能。程序设计的抽象性往往使学生望而生畏,本书选用简单实用的例题,目的是让学生能尽快入门,并对程序设计产生兴趣,能体会到编程和调试的乐趣。

全书共 8 章,第 1、2 章介绍 C 语言的基本概念、基本数据类型和表达式等知识;第 3 章主要介绍程序控制结构,语句的概念、顺序结构、选择结构和循环结构的概念和应用;第 4 章主要介绍数组,一维数组、字符串和二维数组的基本概念和应用;第 5 章主要介绍函数,变量的存储性质,递归函数等;第 6 章主要介绍指针,字符串操作、指针数组、指针函数等;第 7 章主要介绍结构体、共用体、链表和编

译预处理等知识；第 8 章主要介绍文件及其基本应用。附录中给出了常用的 ASCII 码表、库函数、常见错误提示，还给出了全国计算机等级考试二级 C 语言程序设计考试大纲(2013 年版)。

本书的例题程序代码是在 Microsoft Visual C++ 6.0 环境下进行编译运行的。本书及配套实验指导书对在 Microsoft Visual C++ 6.0 环境下如何调试程序给出了实际案例，旨在帮助学生掌握程序调试的技能，这对于学习 C 语言和程序设计都是至关重要的。

本书由多年承担本课程一线教学、经验丰富的多位老师共同编写，由姚立新、梁宏涛担任主编，苏爱玲、林旭平、姜琴、房正华、杨新艳承担不同章节的编写，全书由姚立新统稿。

本书有配套的《C 语言程序设计实验指导》教材，该教材中包括本书各章节的实验案例、实验内容和本书的练习题及课后习题解答。

本书可以作为一般普通高等院校理工科 C 语言程序设计教材，特别适用于以培养应用型人才为目标的高等学校使用，本书也适用于自学 C 语言的读者，还可以作为计算机等级考试的参考用书。

本书配有完整的教学课件教学视频等教学资源，读者可以登录 <http://221.215.217.105:8088/web/cyyxcsj/home> 网站，进行在线学习和下载资料。

由于时间紧促，编者水平有限，本书难免有错误和不足，敬请同行和读者批评指正，以便我们在以后修订时改进。

编者

2013 年 7 月

目 录

第 1 章 C 语言程序设计概述	1
1.1 程序与程序设计语言	1
1.1.1 程序与程序设计	1
1.1.2 程序设计语言	2
1.1.3 算法与数据结构	3
1.2 C 语言概述	3
1.2.1 C 语言的特点	3
1.2.2 C 语言程序的基本结构	4
1.2.3 C 语言的基本要素	5
1.3 设计 C 语言程序的基本过程	6
1.3.1 C 语言程序编程环境	6
1.3.2 编译、连接和运行	8
1.4 如何学习 C 语言	8
习题 1	9
第 2 章 数据类型和表达式	11
2.1 C 语言数据类型	12
2.2 变量	13
2.3 常量	15
2.3.1 整型常量	15
2.3.2 实型常量	16
2.3.3 字符型常量	16
2.3.4 符号常量	17
2.4 数据的输入、输出	18
2.4.1 printf() 函数	18
2.4.2 scanf() 函数	20
2.5 运算符与表达式	23
2.5.1 算术运算符	23
2.5.2 关系运算符	25
2.5.3 逻辑运算符	25
2.5.4 赋值运算符	26
2.5.5 条件运算符	27
2.5.6 逗号运算符	27
2.5.7 位运算符	28

2.6 类型转换	29
2.6.1 自动类型转换	29
2.6.2 强制类型转换	30
习题 2	31
第 3 章 程序控制结构	33
3.1 概述	33
3.2 顺序结构	34
3.3 选择结构	36
3.3.1 单分支结构	36
3.3.2 双分支结构	37
3.3.3 分支结构的嵌套	38
3.3.4 多路分支结构	41
3.4 循环结构	46
3.4.1 for 循环结构	47
3.4.2 while 循环结构	49
3.4.3 do-while 循环结构	50
3.4.4 循环结构的嵌套	52
3.5 break 和 continue 控制语句	53
3.5.1 break 语句	53
3.5.2 continue 语句	55
3.6 程序控制结构的综合应用	56
习题 3	61
第 4 章 数组	64
4.1 一维数组	64
4.1.1 一维数组的定义和引用	64
4.1.2 一维数组的初始化	66
4.1.3 一维数组编程实例	66
4.2 一维字符数组和字符串	74
4.2.1 一维字符数组的定义和初始化	74
4.2.2 字符串概念	75
4.2.3 字符串存储	75
4.2.4 字符串输出	77
4.2.5 字符串的处理	78
4.3 二维数组	81
4.3.1 二维数组的定义和引用	81
4.3.2 二维数组的初始化	82
4.3.3 二维数组编程实例	83
习题 4	87
第 5 章 函数	88
5.1 模块化程序设计	88
5.2 函数的定义和调用	90
5.2.1 函数的定义	90
5.2.2 函数的调用	92

5.3	变量的存储属性	97
5.3.1	自动(auto)变量	98
5.3.2	寄存器(register)变量	99
5.3.3	静态(static)变量	99
5.3.4	用 extern 声明外部变量	101
5.4	函数的嵌套调用	104
5.5	递归函数	105
5.6	数组作函数参数	110
5.6.1	数组元素作函数实参	110
5.6.2	一维数组名作函数参数	111
5.6.3	二维数组名作函数参数	114
	习题 5	117
第 6 章	指针	118
6.1	指针的概念	118
6.2	指针与简单变量	120
6.2.1	指针变量的定义与引用	120
6.2.2	指针变量的特殊性	122
6.2.3	指针变量作为函数的参数	123
6.3	指针与一维数组	127
6.3.1	数组名是一个指针常量	127
6.3.2	指针的运算	129
6.3.3	将数组地址传递给函数	131
6.4	指针与字符串	135
6.4.1	使用指针表示字符串	135
6.4.2	动态内存分配	136
6.4.3	常用的字符串处理函数	138
6.5	指针进阶	144
6.5.1	二级指针	144
6.5.2	指针与二维数组	145
6.5.3	指针数组	145
6.5.4	命令行参数	149
6.5.5	返回指针的函数	151
6.5.6	指向函数的指针	152
	习题 6	154
第 7 章	构造数据类型与编译预处理	156
7.1	结构体	156
7.1.1	结构体类型的定义	157
7.1.2	结构体变量的定义	158
7.1.3	结构体变量的引用	161
7.1.4	结构体变量的初始化	161
7.2	结构体数组	163
7.3	线性链表	167
7.3.1	链表的概念	168

7.3.2	链表的基本操作	169
7.4	共用体	174
7.4.1	共用体类型定义	174
7.4.2	共用体变量的定义、引用	175
7.4.3	共用体变量的赋值	175
7.5	枚举类型	177
7.6	自定义类型名	179
7.7	编译预处理	180
7.7.1	编译预处理命令简介	180
7.7.2	宏定义	181
7.7.3	文件包含	185
7.7.4	条件编译	186
	习题 7	189
第 8 章	文件	192
8.1	文件的基本概念	192
8.2	文件指针	193
8.3	文件的打开、读写与关闭	194
8.3.1	文件的打开	194
8.3.2	文件的关闭	195
8.3.3	文件的读写	195
8.3.4	文件读写函数的选用原则	203
8.4	文件定位	203
	习题 8	205
附录		208
附录 I	ASCII 码表	208
附录 II	C 标准库函数	209
2.1	输入与输出函数<stdio.h>	209
2.2	字符类测试函数<ctype.h>	216
2.3	字符串函数<string.h>	217
2.4	数学函数<math.h>	220
2.5	实用函数<stdlib.h>	222
2.6	诊断函数<assert.h>	224
2.7	变长变元表函数<stdarg.h>	225
2.8	非局部跳转函数<setjmp.h>	225
2.9	信号处理函数<signal.h>	226
2.10	日期与时间函数<time.h>	226
2.11	由实现定义的限制<limits.h>和<float.h>	228
附录 III	C 语言错误提示	230
3.1	致命错误信息	230
3.2	一般错误信息	230
附录 IV	编程风格	236
附录 V	全国计算机等级考试二级C语言程序设计考试大纲(2013年版) ...	238

第 1 章 C 语言程序设计概述

C 语言是面向过程的高级程序设计语言之一，它具有数据类型丰富、灵活高效和结构化等特征。本章主要介绍 C 语言的来历，C 语言的特征和如何使用 VC6.0 设计一个简单的 C 程序，并编译和运行这个 C 程序，最后就如何学好 C 语言给出建议。

知识 结 构

1. 程序与程序设计语言

- ① 程序
- ② 程序设计
- ③ 算法与数据结构
- ④ 程序设计语言的发展

2. C 语言概述

- ① C 语言的特点
- ② C 语言程序的基本结构
- ③ C 语言的基本要素

3. C 程序设计基本过程

- ① C 编程环境
- ② 编辑源程序
- ③ 编译、连接、运行

4. 如何学好 C 语言

1.1 程序与程序设计语言

1.1.1 程序与程序设计

程序是解决特定问题的步骤，计算机程序是用计算机程序设计语言编写的，最终能够在计算机上运行的指令的序列。计算机执行的每一个操作，都是按预先设好的指令完成的。如图 1.1 所示，勇气号火星探测器在火星的行走、对岩石取样等都是一个个具体的任务，都需要设好指令。用指令编写成能够完成一个具体任务的序列，这就是程序。

程序设计是指设计、编制、调试程序的方法和过程。程序设计通常分为问题建模、算法设计、编写代码和编译调试几个阶段。

1.1.2 程序设计语言

语言作为一种交流的工具，其要素有两点：一组符号和一组规则，按照规则由记号构成的记号串的总体就是语言。程序设计语言，简称编程语言，就是人与计算机交流的工具，用来向计算机发出指令，使得计算机能够准确完成解决特定问题设定的步骤。在程序设计语言中，记号串就是程序。程序设计语言的基本成分有：

- ① 数据成分——描述程序处理的对象，在计算机中统称为数据。
- ② 运算成分——描述程序中对数据的处理，即运算。
- ③ 控制成分——描述程序的流程控制。
- ④ 传输成分——描述对数据的传输，即输入和输出。

计算机程序设计语言的发展，经历了从机器语言、汇编语言到高级语言的历程，目前面向对象的程序设计思想和软件开发技术已经成为主流。

1. 机器语言

机器语言是指由二进制编码表示的一种语言。从始至今，计算机内部能识别的只有 0 和 1 构成的二进制代码。最早的程序都是采取二进制的机器语言编写的，编程者要精通计算机的内部结构和熟记 0 和 1 构成的机器指令才能够设计程序，这相当困难，影响了计算机的普及和应用。

2. 汇编语言

为了解决计算机编程难的问题，人们在 1952 年研究了能够代替机器语言的助记符语言，即汇编语言。汇编语言通过助记符代替机器指令，克服了机器语言难记和不易读等缺点，但是汇编语言依赖于特定机器，可移植性差，一般只用在系统软件和特定领域的开发。

3. 面向过程的高级语言

为了普及计算机，需要编制各领域的应用软件，让计算机发挥更大的作用，人们需要更易于掌握的和接近人类自然语言的编程语言，面向过程的高级语言就产生了。计算机不能直接运行使用高级语言编写的程序，需要编译或解释系统把它“翻译”成机器语言程序才能运行。不同的高级语言有不同的编译系统。

高级语言种类很多，有 Fortran、Basic、ALGOL、C 等。面向过程的高级语言主要是面向数学公式和算法的语言。具有丰富的数据类型、结构化的语句、模块化的结构、简单易懂、高效等特点。

4. 面向对象的高级语言

随着编写大型应用软件的需要，20 世纪 80 年代，产生了更接近人类自然交流的语言，如：C++，Perl，Visual Basic，Java 等。计算机程序设计和软件开发全面进入面向对象的程序设计阶段。

随着各种软件系统的设计实现，计算机已经深入到国防安全、工业控制、金融通信、互联网等有关生活的方方面面。



图 1.1 勇气号火星探测器

1.1.3 算法与数据结构

著名的计算机科学家，Pascal 之父——Niklaus·Wirth 曾提出一个公式：

数据结构+算法=程序

他因此获得了计算机界的诺贝尔奖——图灵奖。

1. 算法

算法就是为解决特定问题所采取的方法和步骤，解决不同问题所采用的算法不同。

在程序设计中，算法就是一系列解决问题的清晰指令。如果一个算法有缺陷或者不适合解决某个问题，执行这个算法，问题将得不到解决。解决同一问题可能有不同的算法，效率可能大不一样，衡量算法的优劣指标有时间复杂度和空间复杂度。

算法的 5 大特征：

- ① 有穷性——一个算法必须保证执行有限步之后结束。
- ② 确切性——算法的每一步骤必须有确切的定义。
- ③ 可行性——原则上算法能够精确地运行。
- ④ 零个或多个输入——一个算法可以有零个或多个输入。
- ⑤ 一个或多个输出——一个算法必须有输出，没有输出的算法毫无意义。

2. 数据结构

数据结构即计算机存储、组织数据的方式。数据结构是指相互之间存在一种或几种特定关系的数据的集合。数据结构与算法密不可分，一个良好的数据结构可以简化算法，反过来，明确了问题的算法，才能较好地设计数据结构，两者相辅相成。通常情况下，精心选择的数据结构可以带来更高的运行效率和存储效率。

1.2 C 语言概述

1.2.1 C 语言的特点

C 语言是近些年来最流行的计算机高级语言，根据 TIOBE 的统计，C 语言的用户占有量在 Java 出现之前一直排在第一位。直到 2000 年 Java 出现后 C 语言才有了对手。如图 1.2 所示，2010 年 11 月以来 C 语言又多次回到第一的位置。这主要是因为 C 语言既可以用来编写系统软件，也可以用来编写应用软件。C 语言的主要特点如下。

① 语言简洁、紧凑，使用方便灵活。C 语言一共有 32 个关键字，9 种控制语句，程序书写形式自由，语法限制少。

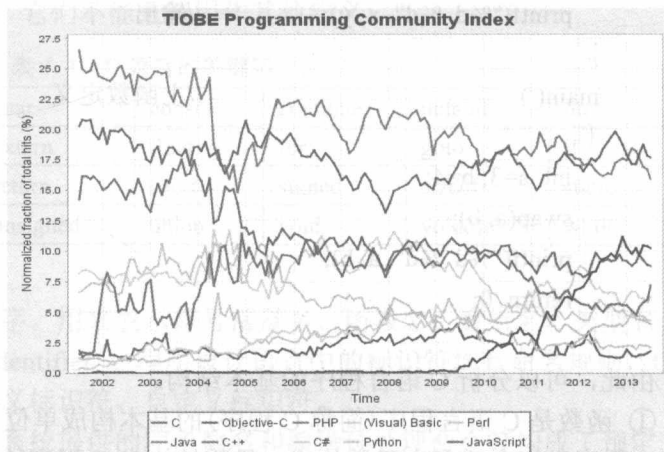


图 1.2 各种语言用户比例统计图

② 运算符丰富、数据处理能力强。C 语言的运算符包含的范围很广，共有 34 种运算符。

③ 数据结构丰富，具有现代高级语言提供的大多数数据结构。C 语言的数据类型有整型、实型、字符型、数组类型、指针类型、结构体类型等。C 语言能用来实现各种复杂的数据结构的运算，尤其是指针型数据，使用起来比较灵活、多样。

④ C 语言是一种结构化的程序设计语言。C 语言具有结构化的控制语句，并用函数作为程序的模块单位，是一种理想的结构化的编程语言。

⑤ C 语言允许直接访问物理地址，能进行位(bit)操作，实现汇编语言的大部分功能，还可以直接对硬件进行操作。

⑥ 可移植性好(与汇编语言相比)。源代码基本上不做修改，经过重新编译，就能够用于各种操作系统。

当然，C 语言也有其不足之处。例如，C 语言的语法限制不太严格，在增加程序设计的灵活性的同时，在一定程度上降低了某些安全性，这对程序设计人员提出了更高的要求。

由于开发大型软件的需要，近年来，面向对象的 C++ 语言、Java 语言得到进一步推广，但是不应该得出“C 语言已经过时了，不必学习 C 语言了”的结论。不应该把面向对象的程序设计与面向过程的程序设计对立起来，面向对象的基础是面向过程。自 20 世纪 90 年代以来，我国学习使用 C 语言的人越来越多，学习 C 语言不仅能培养我们的编程能力，同时还能使我们养成良好的逻辑思维习惯，培养我们分析问题和解决问题的能力。

1.2.2 C 语言程序的基本结构

【例 1.1】一个 C 语言程序示例。

```

/*****
/*   C 语言程序实例   */
*****/

#include<stdio.h>           //编译预处理命令
void swap(int x,int y)      //函数定义，函数首部
{                             //函数体
    int t;                  //定义变量
    t=x; x=y; y=t;
    printf("%d %d", x, y);  //输出
}

main()                      //主函数定义
{
    int a=3, b=4;
    swap(a, b);
    printf("%d %d", a, b);
    return 0;
}

```

由此，可以分析 C 语言程序的基本结构。

① 函数是 C 语言程序(简称 C 程序)的基本构成单位。

C 程序由一个或多个函数构成，函数是构成 C 程序的基本单位。一个 C 程序中必须包含一个主函数 main()，而且最多只能有一个主函数。除了一个主函数外还可以有零个或多个其

他函数，如 swap() 函数。

② 一个 C 程序总是从 main() 函数开始执行，至 main() 函数的最后一句结束，函数中可以调用其他函数。

③ 函数的一般结构：

函数由函数首部加函数体构成，结构如下：

返回值类型 函数名(参数列表)

{

函数体

}

返回值类型可以缺省，如 main()，缺省时表示返回整型值，等价于 int main()。参数列表可以没有，表示没有参数，也可以写成 main(void)，但函数名后的一对圆括号不能省略，这是函数的标志。函数体紧跟函数首部，以左大括号“{”开始，以右大括号“}”结束。通常包括变量说明部分和执行语句部分，说明部分必须在执行部分之前。

C 语言中函数分两种，自定义函数和库函数，库函数是 C 编译系统提供的，如标准输入输出函数 scanf() 和 printf() 等，它们的定义存储在“stdio.h”文件中。因此当要调用 printf()、scanf() 等函数时需要在程序的开头加上一条编译预处理命令#include<stdio.h>。

④ 执行语句以分号结尾，分号是语句的一部分，表示一条语句的结束。

⑤ 注释部分是“/*”和“*/”括起来的部分，它对程序的功能或含义加以说明，编译时不起作用。注释部分可以出现在程序任何位置，单行注释也可以用“//”开头。

⑥ C 语言本身没有输入输出语句。由于输入输出操作涉及具体的计算机设备，因此 C 语言对输入输出实行“函数化”，把输入输出操作放在函数中处理。输入和输出的操作由库函数完成，如 scanf() 和 printf()。

建议用 C 语言编程时，每行写一条语句，注意缩进、留空、对齐、加注释，养成良好的编程风格，使得程序结构清晰，可读性强，方便排错、调试、修改。

1.2.3 C语言的基本要素

1. 关键字

关键字是 C 语言规定的具有特定意义的字符串。一共有 32 个关键字，由系统定义，如表 1.1 所示。关键字又称为保留字，它们不能重新用作其他定义。

表 1.1 C 语言的关键字

auto	break	case	char	const	continue	default	do
double	else	enum	extern	float	for	goto	if
int	long	register	return	short	signed	sizeof	static
struct	switch	typedef	unsigned	union	void	volatile	while

2. 标识符

简单地说，标识符就是一个名字。用来表示符号常量名、函数名、数组名、类型名、文件名、有效字符序列称为标识符(identifier)。程序设计语言中的标识符均有命名规则，C 语言中标识符有 3 类：关键字、预定义标识符、自定义标识符。

① 预定义标识符：C 语言编译系统提供的库函数名和编译预处理命令等构成了预定义标识符。

② 自定义标识符：用户可自定义标识符，C 语言标识符命名规则规定，标识符只能由字母、数字和下划线三种字符组成，且第一个字符必须是字母或下划线。

如下面列出的几个字串就是合法的标识符：

Sum average _total class day Month
student_name tan lotus_1_2_3 basic li_ling

下面列出的几个字串不是合法的标识符：

M. d.j ohn, y 123 # 33 3d64 a>b

C 语言区分大小写字母。因此，A 和 a，I 和 i，Sum 和 sum 分别表示两个不同的标识符。

标识符的命名应该做到见名知义，如 max，min，sum，count 等，另外要符合业内命名习惯，如变量名一般用小写字母，符号常量名用大写字母，i，j，k 通常用作循环控制变量名。

3. 库函数

C 语言的库函数是系统预先定义好的由编译系统提供的函数，包含了许多通用的常用的功能模块，库函数的引入，极大方便了程序设计人员的使用，也扩大了语言本身的功能。

库函数按照功能分为若干类，程序中如果需要用到库函数，应当在程序开始处用编译预处理命令将包含有库函数定义的相关头文件包含进来。

如要使用输入输出类函数就要用

```
#include<stdio.h>
```

使用数学类函数就要用

```
#include<math.h>
```

各类库函数请参阅附录。

1.3 设计 C 语言程序的基本过程

1.3.1 C 语言程序编程环境

高级语言编写的程序文件称为源代码文件，C 语言编写的源程序文件的扩展名为.C，可以采用各种编辑器编辑 C 语言源程序，如记事本、UltraEdit 等，也可以使用 C 语言编译器的 IDE 环境中自带的编辑器。计算机不能直接识别高级语言程序，要经过编译器编译成二进制的目标代码，生成扩展名为.obj 的文件，然后与相关的库函数连接装配成可执行的代码，生成.exe 文件后才能在计算机上运行。C 语言程序设计的基本过程如图 1.3 所示。

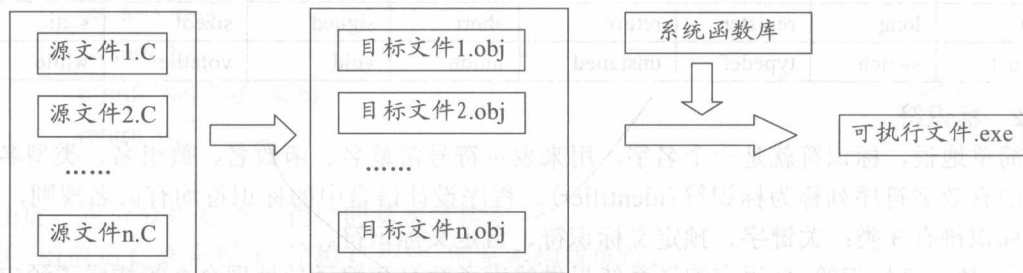


图 1.3 C 程序设计基本过程

现在用 VC++6.0 集成开发工具设计一个 C 语言源程序，然后对其编译、连接，最后运行，查看程序的运行效果。

【例 1.2】写一个最简单的 C 语言程序，在屏幕上输出一个字符串“Hello World!”。

第 1 步：建立源文件，如图 1.4 所示。

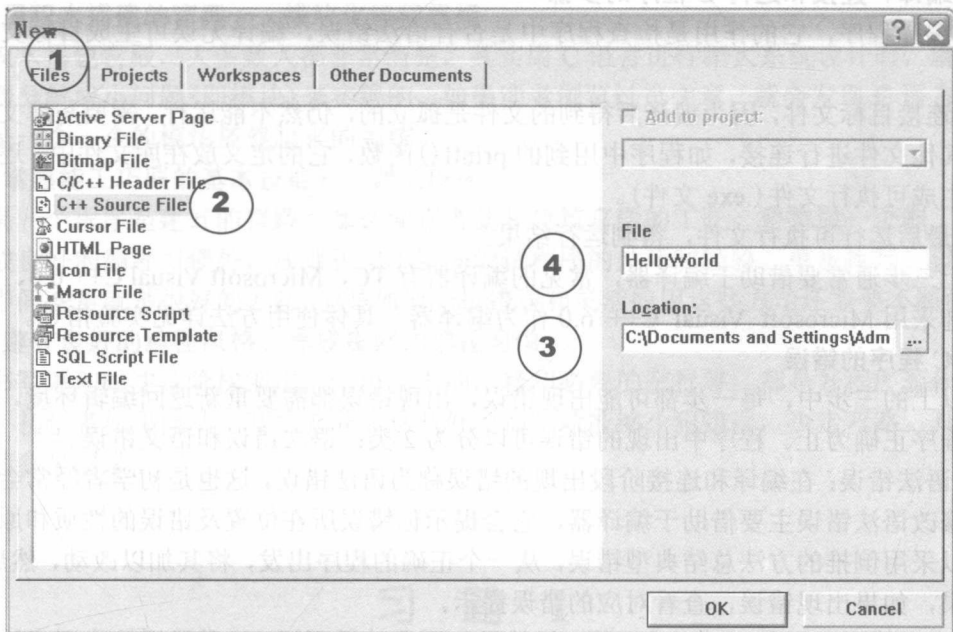


图 1.4 新建源文件窗口

- ① 选 Files(新建文件)标签。
- ② 选文件类型为“C++ Source File”。
- ③ 调整文件存取路径 Location。
- ④ 填写文件名(默认文件扩展名 CPP，可以加.c)。

第 2 步：编辑源文件，如图 1.5 所示。

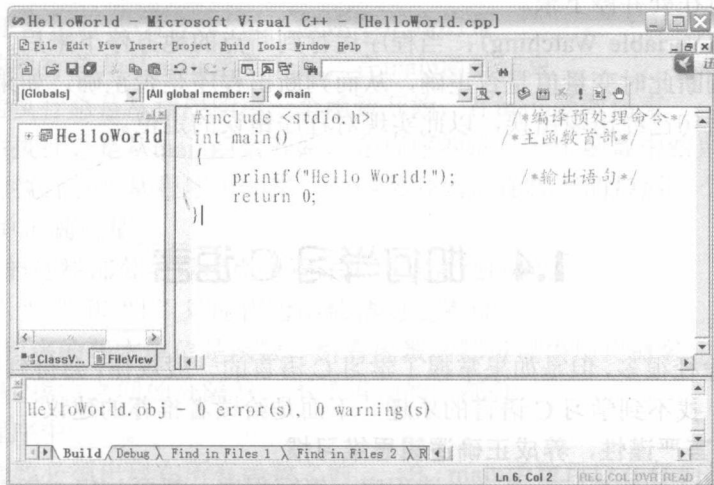


图 1.5 源文件编辑窗口

1.3.2 编译、连接和运行

C 语言程序是否正确的唯一标准就是程序的运行结果与预期结果是否完全相同，在运行程序之前必须先对程序进行编译、连接。

1. 编译、连接和运行 C 程序的步骤

① 编译程序，它的作用是检查程序中是否有语法错误，编译无误可生成目标文件(.obj 文件)。

② 连接目标文件，因为编译后得到的文件是孤立的，仍然不能运行，需要与库文件或程序中的其他文件进行连接，如程序中用到的 `printf()` 函数，它的定义放在库文件中。连接无误后便可生成可执行文件(.exe 文件)。

③ 最后运行可执行文件，得到运行结果。

以上三步通常要借助于编译器，常见的编译器有 TC、Microsoft Visual C++ 6.0、gcc 等。本书全部采用 Microsoft Visual C++ 6.0 作为编译器，具体使用方法详见实验指导。

2. C 程序的错误

在以上的三步中，每一步都可能出现错误，出现错误都需要重新返回编辑环境，修改程序直到程序正确为止。程序中出现的错误可以分为 2 类：语法错误和语义错误。

① 语法错误：在编译和连接阶段出现的错误称为语法错误，这也是初学者经常会遇到的情况，修改语法错误主要借助于编译器，它会提示你错误所在位置及错误的性质和原因。

可以采用倒推的方法总结典型错误，从一个正确的程序出发，将其加以改动，然后编译，查看结果，如果出现错误，查看对应的错误提示。

② 语义错误：程序可以正常运行，但结果与预期结果不同，这种错误称为语义错误或逻辑错误。语义错误的修改主要借助于调试。

3. 调试程序

调试的主要方法有单步跟踪和设置断点两种，这两种方法都要借助于观察变量。调试的原理就是通过对程序中语句的逐条执行或者多条语句一次性执行，对程序错误进行定位。

① 单步跟踪(Trace Step by Step)：即一步一步跟踪程序的执行过程。

② 设置断点(Break Point Setting)：可以在程序的任何一个语句上做断点标记，程序会直接运行到断点所在处并停下来。

③ 观察变量(Variable Watching)：当程序运行到断点的地方停下来后，就可以观察各个相关变量的值，判断此时变量值是否正确，从而判断出程序是否正确，如果发生错误，说明在该处或之前的语句已经出现错误，以此实现对程序错误的定位。

1.4 如何学习 C 语言

学习 C 语言途径很多，但是如果掌握了学习 C 语言的一般规律，会得到事半功倍的效果，不至于越学越累，找不到学习 C 语言的乐趣。下面是给读者推荐的建议。

1. 体会 C 语言严谨性，养成正确逻辑思维习惯

严谨性是 C 语言的重要特征之一。读程序、编写程序，都要一句一句来，注意对程序进行分析，加强逻辑思维锻炼，并以流程图的方式描述出分析的结果，即解决问题的流程。流