

21世纪全国高等院校实用规划教材



SHIFT

ENTER

C语言程序设计教程

杨忠宝 王晶宝 主编

教材预览、申请样书



微信公众号: pup6book



北京大学出版社
PEKING UNIVERSITY PRESS

21 世纪全国高等院校实用规划教材

C 语言程序设计教程

主 编 杨忠宝 王晶莹
副主编 任长虹 董晓明
黄 亮 于大伟



北京大学出版社
PEKING UNIVERSITY PRESS

内 容 简 介

本书是学习 C 语言程序设计的基础教材。本书的特点是在内容安排上采用循序渐进的方式,在组织形式上采用通俗易懂的案例教学和启发式教学的方式,并辅以大量的便于说明问题的案例,用案例带动知识点的方法进行讲解,以一节为一个单元,对知识点进行了细致的取舍和编排,按小节细化知识点并结合知识点介绍了相关的实例,将知识和案例放在同一节中,知识和案例相结合。

本书可作为高校各专业 C 语言教材和全国计算机等级考试参考书,也可供对 C 语言感兴趣的其他读者自学使用。

图书在版编目(CIP)数据

C 语言程序设计教程/杨忠宝,王晶莹主编. —北京:北京大学出版社,2015.8

(21 世纪全国高等院校实用规划教材)

ISBN 978-7-301-25712-8

I. ①C… II. ①杨…②王… 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(2015)第 084303 号

书 名	C 语言程序设计教程
著作责任者	杨忠宝 王晶莹 主编
责任编辑	郑 双
标准书号	ISBN 978-7-301-25712-8
出版发行	北京大学出版社
地 址	北京市海淀区成府路 205 号 100871
网 址	http://www.pup.cn 新浪微博:@北京大学出版社
电子信箱	pup_6@163.com
电 话	邮购部 62752015 发行部 62750672 编辑部 62750667
印 刷 者	北京富生印刷厂
经 销 者	新华书店
	787 毫米×1092 毫米 16 开本 18.5 印张 435 千字
	2015 年 8 月第 1 版 2015 年 8 月第 1 次印刷
定 价	39.00 元

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

举报电话:010-62752024 电子信箱:fd@pup.pku.edu.cn

图书如有印装质量问题,请与出版部联系,电话:010-62756370

前 言

C 语言在计算机程序设计领域应用非常广泛，它具有功能丰富、语句简洁、使用方便、语法灵活、数据结构多样、能操作硬件、高移植性和通用性等诸多优点。C 语言既有高级语言的特点，又有汇编语言等低级语言的特点，已经成为编制系统软件和应用软件的首选语言。

C 语言是我国各高校普遍开设的一门重要的计算机基础课程，也是计算机专业学生学习程序设计语言的必修课程，同时还是全国计算机等级考试二级考试的主要语言。通过本课程的学习，能使学生应用计算机解决实际问题的能力得到进一步的提高，为后续的计算机应用课程打下坚实的基础。本书内容主要是根据全国计算机等级考试二级考试大纲设计的，所以本书既可作为各大学 C 语言课程的教学用书，也可以作为国家计算机二级考试的参考用书。

编者结合多年从事 C 语言及计算机二级教学的经验编写此书，本书主要特点如下。

(1) 分析近几年全国计算机二级考试试题，将知识点、考点合理分布在各章，各章节内容由浅入深，前后内容衔接合理。

(2) 理论联系实际，尽可能将概念、知识点与例题结合起来，力求通俗易懂。

(3) 书中大部分例题和习题都是选自历年国家计算机二级考试原题。大部分例题都是经典题型，适合举一反三。课后习题的题型与国家计算机二级考试题型完全一致。

(4) 每道例题都添加了必要的中文注释，并且程序中输入/输出提示信息也都采用中文，增加了程序的可读性。

(5) 稍有难度的例题都配有算法分析、设计步骤、配套图表及运行结果。

(6) 流程控制语句均配有程序流程图。

(7) 书中提供的例题均在 Visual C++6.0 平台下实际运行通过。

本书主要内容：C 语言的发展历史、特点、程序结构以及程序开发过程；各种数据类型、常量和变量、各种类型的运算符和表达式及表达式的求值过程；几种常用的顺序执行语句、常用的输入/输出函数；选择结构程序设计；循环结构程序设计；数组的定义、引用和初始化方法，常用的排序方法，字符数组；函数的定义、说明和调用方法，函数间参数传递，全局变量和局部变量，变量的存储类别；指针的概念、指针变量定义和初始化、指针运算符、指针和字符串的关系、指针和数组的关系、指针和函数的关系及二级指针；结构体类型和结构体变量定义方法、结构体数组和结构体指针、单向链表；文件的概念、打开和关闭方法、文件读写方法及文件位置指针定位方法；宏定义、文件包括和条件编译等编译预处理命令；常用位运算符等。

为了便于教学和自学，我们还编写了以下与本书配套的教学工具：

《C 语言程序设计实验教程》：包括实验指导、实验项目、课程设计、自测练习；

《C 语言程序设计教程》教材的配套 PPT 课件；

《C 语言程序设计教程》教材的源程序。源程序已在 VC++6.0 环境下编译调试通过。上述课件和源程序可从 www.pup6.cn 下载。

本书由杨忠宝、王晶莹、任长虹、董晓明、黄亮、于大伟编写。全书由杨忠宝统稿。由于编者水平有限，书中难免存在不足之处，殷切希望读者批评指正。

编者

2015 年 1 月

目 录

第 1 章 C 语言概述	1
1.1 C 语言的历史	1
1.2 C 语言的特点	2
1.3 C 语言程序的开发过程	3
1.3.1 C 语言调试步骤	3
1.3.2 Visual C++ 6.0 环境下调试 程序方法	4
1.4 简单的 C 语言程序	10
习题	12
第 2 章 数据类型、运算符和表达式	14
2.1 数据类型	14
2.2 标识符、常量与变量	15
2.3 整型数据	18
2.4 实型数据	21
2.5 字符型数据	23
2.6 基本运算符和表达式	26
2.6.1 算术运算符和算术表达式	27
2.6.2 赋值运算符和赋值表达式	29
2.6.3 逗号运算符和逗号表达式	30
2.6.4 求字节数运算符	30
2.7 类型转换	31
习题	33
第 3 章 顺序结构程序设计	36
3.1 结构化程序设计	36
3.2 C 语言的语句	38
3.3 数据的输出	39
3.3.1 格式输出函数——printf 函数	39
3.3.2 字符输出函数——putchar 函数	44
3.4 数据的输入	44
3.4.1 格式输入函数——scanf 函数	44
3.4.2 字符输入函数——getchar 函数	47
3.5 顺序结构程序举例	48
习题	50
第 4 章 选择结构程序设计	53
4.1 关系运算符、逻辑运算符、条件 运算符	53
4.1.1 关系运算符和关系 表达式	53
4.1.2 逻辑运算符和逻辑 表达式	54
4.1.3 条件运算符和条件表达式	56
4.2 选择结构程序设计	57
4.2.1 if 语句	57
4.2.2 switch 语句	66
4.3 选择结构程序设计举例	69
习题	73
第 5 章 循环结构程序设计	76
5.1 while 语句	76
5.2 do-while 语句	79
5.3 for 语句	81
5.4 break 语句和 continue 语句	84
5.4.1 break 语句	84
5.4.2 continue 语句	85
5.5 循环嵌套	86
5.6 程序举例	89
习题	93
第 6 章 数组	98
6.1 一维数组	98
6.1.1 一维数组的定义和引用	98
6.1.2 一维数组的初始化	102
6.1.3 一维数组程序举例	103

6.2	二维数组	109	8.2	指针变量的定义和引用	160
6.2.1	二维数组的定义和引用	109	8.2.1	指针变量的定义	160
6.2.2	二维数组的初始化	110	8.2.2	指针变量的引用	161
6.2.3	二维数组程序举例	113	8.2.3	指针变量的算术运算	164
6.3	字符数组与字符串	115	8.2.4	指针变量作为函数的参数	165
6.3.1	字符数组的定义和初始化	115	8.3	指针与一维数组	167
6.3.2	字符串	117	8.3.1	通过指针变量引用数组 元素	167
6.3.3	字符串处理函数	122	8.3.2	用数组名及指针作为 函数的参数	169
6.3.4	程序举例	124	8.4	指针与二维数组	173
习题		127	8.4.1	二维数组的地址	173
第7章	函数	132	8.4.2	指向二维数组的指针变量	176
7.1	函数概述	132	8.5	指针与字符串	177
7.2	函数定义	134	8.5.1	字符串的表现形式及访问 方式	177
7.2.1	函数定义的一般形式	134	8.5.2	使用字符数组和字符型 指针变量处理字符串的区别	181
7.2.2	函数的返回值	135	8.6	指针与函数	183
7.3	函数调用	136	8.6.1	返回指针值的函数	183
7.3.1	函数调用的一般形式	136	8.6.2	指向函数的指针	184
7.3.2	对被调函数的声明	137	8.7	二级指针和指针数组	185
7.3.3	参数传递	138	8.7.1	二级指针	185
7.4	数组作函数参数	139	8.7.2	指针数组	186
7.4.1	数组元素作函数实参	139	8.7.3	main 函数的参数	189
7.4.2	数组名作函数参数	140	习题		191
7.5	函数的嵌套调用	142	第9章	结构体与链表	196
7.6	函数的递归调用	143	9.1	结构体类型变量的定义	196
7.7	局部变量和全局变量	146	9.2	结构体类型变量的引用	199
7.7.1	局部变量	146	9.3	结构体的初始化	201
7.7.2	全局变量	147	9.4	结构体与数组	203
7.8	变量的存储类别	148	9.4.1	结构体中包含数组	203
7.8.1	静态存储方式和动态存储 方式	148	9.4.2	结构体数组	203
7.8.2	变量的存储类别	149	9.5	结构体和指针	206
习题		153	9.5.1	结构体中包含指针	206
第8章	指针	159	9.5.2	指向结构体的指针	208
8.1	指针的基本概念	159	9.6	用结构体指针处理链表	212
8.1.1	变量与地址	159			
8.1.2	指针与指针变量	160			
8.1.3	直接访问与间接访问	160			

9.6.1	链表介绍	212	10.5.2	fseek 函数	254
9.6.2	动态存储分配	213	习题		256
9.6.3	链表的基本操作	215	第 11 章 编译预处理		259
9.7	共用体	228	11.1	宏定义	259
9.8	枚举	231	11.1.1	无参数的宏定义	260
9.8.1	枚举类型的定义和枚举变量的定义	231	11.1.2	带参数的宏定义	262
9.8.2	枚举变量的使用	232	11.2	文件包含	264
9.9	类型定义	234	11.3	条件编译	267
9.10	应用举例	236	11.3.1	#if 命令	267
习题		238	11.3.2	#ifdef ... #else ... #endif	267
第 10 章 文件		241	11.3.3	#ifndef ... #else ... #endif	269
10.1	文件概述	241	11.3.4	#undef	269
10.2	文件类型指针	243	11.3.5	应用举例	270
10.3	文件的打开与关闭	243	习题		271
10.3.1	文件打开函数 fopen	243	第 12 章 位运算		273
10.3.2	文件关闭函数 fclose	245	12.1	二进制位逻辑运算	273
10.4	文件的读写操作	245	12.1.1	“按位与”运算符 &	274
10.4.1	字符读写函数: fputc 和 fgetc	245	12.1.2	“按位或”运算符	275
10.4.2	格式化读写函数: fprintf 和 fscanf	248	12.1.3	“按位异或”运算符 ^	276
10.4.3	数据块读写函数: fwrite 和 fread	250	12.1.4	“按位取反”运算符 ~	277
10.4.4	字符串读写函数: fputs 和 fgets	252	12.2	移位运算	277
10.5	文件定位函数	253	12.2.1	左移运算符 <<	277
10.5.1	rewind 函数	253	12.2.2	右移运算符 >>	278
			习题		279
			附录		281
			参考文献		288

第1章

C语言概述

1.1 C语言的历史

C语言是世界上最为流行的计算机高级语言之一，它设计精巧，功能齐全，既可以用来编写系统软件，也可以用来编写应用软件。C语言的发展历程主要包括诞生、发展和成熟三个阶段。

1. C语言的诞生

C语言的原型是ALGOL60语言。1963年，英国剑桥大学在ALGOL60语言基础上增添了处理硬件的能力，发展成为CPL(Combined Programming Language)。CPL由于规模大，学习和掌握困难，没有流行开来。

1967年，英国剑桥大学的Martin Richards对CPL进行了简化，于是产生了BCPL(Basic Combined Programming Language)。

1970年，美国贝尔实验室的Ken Thompson将BCPL进一步简化，突出了处理硬件的能力，并取名BCPL的第一个字母B作为新语言名称，并且他用B语言写了第一个可在PDP-7机上运行的UNIX操作系统。

但是B语言过于简单，功能有限。于是在1972年，美国贝尔实验室的D. M. Ritchie对B语言进行了完善和扩充，最终设计出了一种新的语言，他取了BCPL的第二个字母C作为这种语言的名字，这就是C语言的诞生。C语言既保持了B语言的精练、接近硬件等优点，又克服了它们过于简单、数据无类型等缺点。

2. C语言的发展

1973年，K. Thompson和D. M. Ritchie两人合作把第5版UNIX的90%代码改用C语言编写。后来，C语言作了多次改进。直到1975年公布第6版UNIX后，C语言的突出优点才引起人们普遍注意。随着UNIX的日益广泛使用，C语言也迅速得到推广。C语言和UNIX可以说是一对孪生兄弟，在发展过程中相辅相成。

3. C 语言的成熟

随着计算机的日益普及,出现了许多 C 语言版本。由于没有统一的标准,使得这些 C 语言之间出现了不一致的地方。为此,美国国家标准化协会(ANSI)为 C 语言制定了一套 ANSI 标准,被称为 87 ANSI C,是现行的 C 语言标准。

1990 年,国际标准化组织 ISO(International Standard Organization)接受了 87 ANSI C 为 ISO C 的标准。1994 年,ISO 修订了 C 语言的标准。

目前,C 语言已先后移植到大、中、小、微型机上。现在 C 语言已风靡全世界,成为世界上应用最广泛的计算机语言之一。并且它已经不依赖于 UNIX 操作系统而独立存在。

目前,最流行的 C 语言编译系统有以下几种版本:Microsoft C, Turbo C, VC++ 6.0 等,这些 C 语言编译系统大多是以 ANSI C 为基础进行开发的,但不同版本的 C 编译系统所实现的语言功能和语法规则略有差别。本书使用的是 VC++ 6.0 编译系统。

1.2 C 语言的特点

C 语言之所以长盛不衰,成为目前世界上使用最广泛的高级语言之一,完全是由于其特点决定的。C 语言主要有以下特点:

(1) C 语言结构简洁、紧凑,使用方便、灵活。C 语言只有 32 个关键字(即保留字),9 种控制语句,压缩了一切不必要的成分,因此 C 语言编写的源程序短,对于语言本身的描述也简单,易于学习、理解和使用。

(2) C 语言运算符丰富。共有 34 种运算符。C 语言把括号、赋值、逗号等都作为运算符处理,从而使其运算类型极为丰富,可以实现其他高级语言难以实现的运算。

(3) C 语言数据类型丰富。C 语言中数据类型可分为基本类型、构造类型、指针类型和空类型。基本数据类型包括:int(整型)、char(字符型)、float(单精度浮点型)、double(双精度浮点型)等;构造类型包括:数组、结构体、共用体和枚举等;指针类型使用十分灵活,用它可以构成链表、树和栈等。指针可以指向各种类型的简单变量、数组、结构体以及函数等。指针在 C 语言中占有重要的地位,是 C 语言区别于其他高级语言的精髓所在。

(4) C 语言提供了丰富的各种类型库函数(标准函数),供用户使用。

(5) C 语言语法限制不太严格,程序设计自由度大。程序书写自由,主要用小写字母表示。一个语句可以写在几行,一行也可以写几个语句。

(6) C 语言提供了汇编语言的大部分功能,允许直接访问物理地址,能进行位操作,可以直接对硬件进行操作。

(7) C 语言是一种结构化程序设计语言,特别适合大型程序的模块化设计。C 语言具有编写结构化程序所必需的基本流程控制语句。C 语言程序是由函数集合构成的,函数各自独立作为模块化设计的基本单位。它所包含的源文件可以分割成多个源程序,分别对其进行编译,然后连接起来构成可执行的目标文件。

(8) C语言生成目标代码质量高,可移植性好。在C语言的语句中,没有依赖于硬件的输入/输出语句,程序的输入/输出功能是通过调用输入/输出函数实现的,而这些库函数是由系统提供的独立于C语言的程序模块,从而便于在硬件结构不同的计算机和各种不同的操作系统之间实现程序的移植。

综上所述,C语言是一种功能很强的语言。但是,它也有一些不足之处:C语言语法限制不严,虽然熟练的程序员编程灵活,但安全性低;运算符丰富,功能强,但难记难掌握。所以,学习C语言不妨先学基本部分,先用起来,熟练后再学习语法规则,进而全面掌握C语言。

总之,由于C语言的上述特点,使得C语言越来越受到程序设计人员的重视,并且已经在广泛的领域里得到了应用。

1.3 C语言程序的开发过程

1.3.1 C语言调试步骤

开发一个C语言程序的基本过程如图1.1所示。

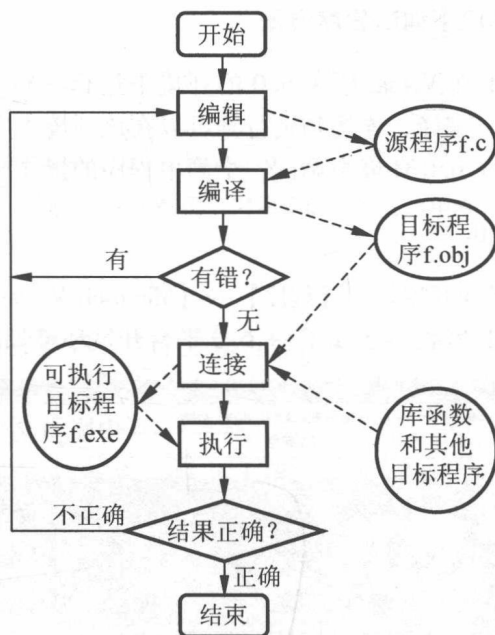


图 1.1 C 程序调试过程示意图

1. 编辑

选择适当的编辑程序,将C语言源程序通过键盘输入到计算机中,并以文件形式存入

磁盘。经过编辑后得到的源程序文件扩展名是 .c 或 .cpp。注意：在 Visual C++ 6.0 下新建的源文件扩展名默认为 .cpp。

2. 编译

通过编辑程序将源程序输入计算机后，需要经过 C 语言编译器将其生成目标程序。在对源程序的编译过程中，可能会发现程序中的一些语法错误，这时就需要重新利用编辑程序来修改源程序，然后再重新编译。经过编译后得到的目标文件扩展名是 .obj。

3. 连接

经过编译后生成的目标文件是不能直接执行的，它需要经过将目标代码和各种库函数连接之后才能生成可执行的目标代码。连接后得到的可执行文件扩展名是 .exe。

4. 执行

经过编译、连接源程序文件就可以生成可执行文件，这时就可以执行了。在 Windows 系统下，在运行窗口中只要键入可执行的文件名，并按 Enter 键，就可执行文件，或双击可执行文件。

1.3.2 Visual C++ 6.0 环境下调试程序方法

C 语言源程序完全能够在 Visual C++ 6.0 的环境下运行。Visual C++ 6.0 提供了全屏幕的程序调试环境，编辑、编译、连接和运行都可以在该环境中完成。

下面介绍在 Visual C++ 6.0 环境中调试一个简单程序的操作步骤。

1. 启动 Visual C++ 6.0

在 Windows 环境下选择【开始】→【程序】→【Microsoft Visual Studio 6.0】→【Microsoft Visual C++ 6.0】命令。启动后，Visual C++ 6.0 平台开发环境如图 1.2 所示。

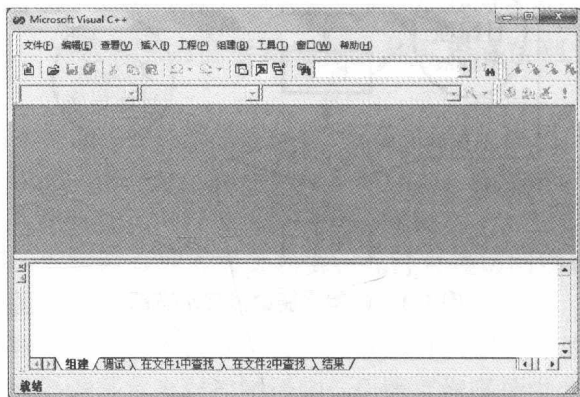


图 1.2 Visual C++ 6.0 平台

2. 建立一个新的工作区

工作区即工作空间，一个工作区可以包含一个或几个工程（一个工程对应一个开发项目）。每个工作区对应一个文件夹，文件夹的名字就是该工作区的名字。该工作区中的所有工程都将存储在该工作区所对应的文件夹中。

选择【文件】→【新建】命令，弹出【新建】对话框。在对话框中选择【工作区】选项卡，然后在工作空间名称文本框当中输入要建立的工作区名称，如 workspace1，如图 1.3 所示。并在“位置[C]”处选择该工作区所在文件夹，图 1.3 中所选择的文件夹是 C 盘根目录，即 C:\，单击【确定】按钮，这样系统就在 C 盘根目录下以工作区名称 workspace1 建立一个文件夹，以后在该工作区建立的所有工程和程序文件都将存储在该工作区文件夹中。

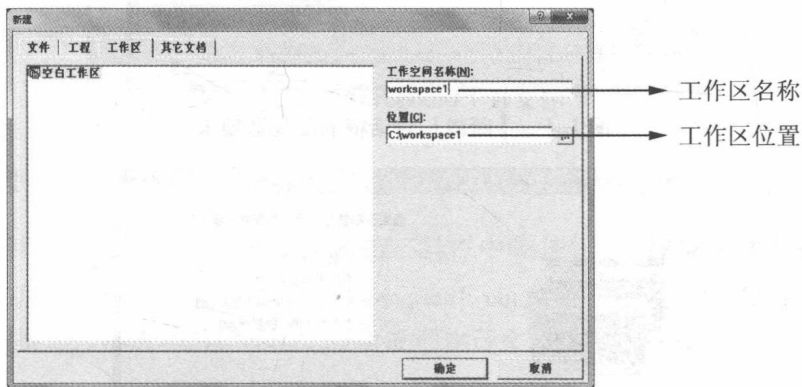


图 1.3 【新建】对话框的工作区选项卡

3. 建立一个新的工程

一个工程对应一个开发项目，多个工程可以保存在同一个工作区中。系统会为每个工程在工作区文件夹下创建一个文件夹，文件夹的名字即为该工程的名字。该工程包含的所有文件都将保存在该工程文件夹中。

选择【文件】→【新建】命令，弹出【新建】对话框，在此对话框中选择【工程】选项卡，在所列工程中选择“Win32 Console Application”（Windows 控制台应用程序）选项，在右面工程名称文本框中输入要建立的工程名称（如 project1）；然后选中“添加到当前工作空间”单选按钮，单击【确定】按钮，如图 1.4 所示。系统弹出如图 1.5 所示的对话框，在该对话框中选中“一个空工程”单选按钮，表示建立空工程，单击【完成】按钮；在新工程信息对话框中单击【确定】按钮，完成新工程的建立。

4. 建立源文件

新建的工程是空的工程，没有具体内容。我们需要在新的工程中创建 C++ 源程序文件，其方法是：单击【文件】菜单，在弹出的子菜单中选择【新建】命令，选择【文件】

选项卡，如图 1.6 所示，选择“C++Source File”选项，同时在右边勾选“添加到工程 [A]”，并选择前面刚建好的工程名“project1”。在“文件名 [N]”下方的文本框中输入一个源文件名，如 program1，单击【确定】按钮。这样就在工程文件夹 project1 中建立了一个源文件，如 program1.cpp，当然同时也建立了几个相关的辅助文件。

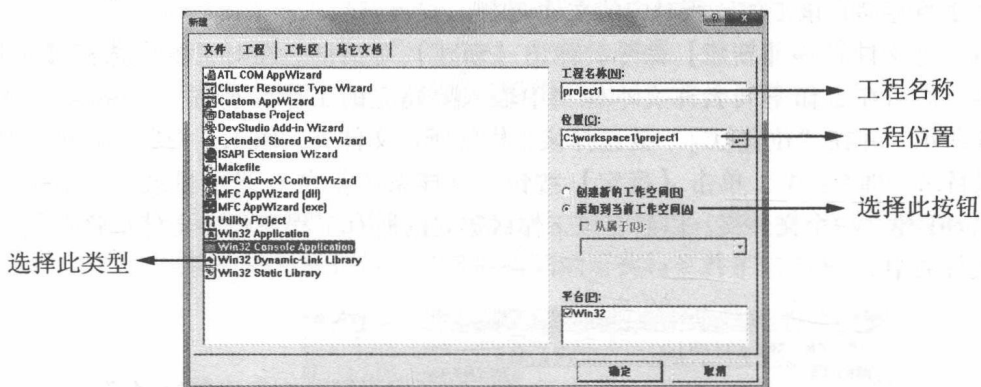


图 1.4 【新建】对话框的工程选项卡

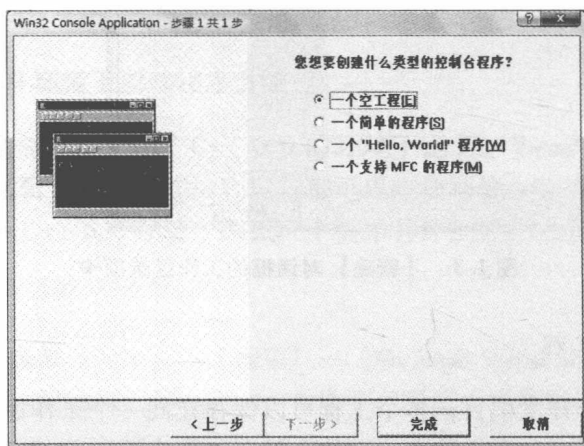


图 1.5 选择工程类型对话框

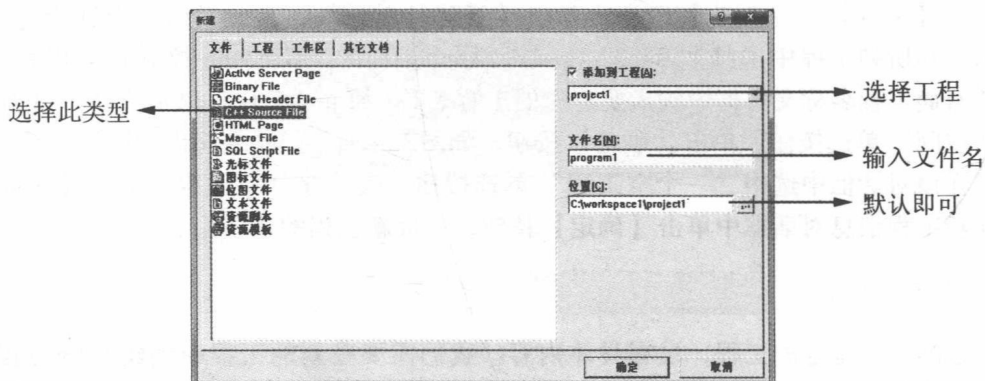


图 1.6 【新建】对话框的文件选项卡

5. 编辑 C 的源程序文件




在编辑区中写入内容，如图 1.7 所示，单击工具栏中保存  按钮，保存文件。




图 1.7 编辑“program1”源文件

6. 连编应用程序

输入源程序代码后即可进行连编，先单击工具栏中  按钮（或按快捷键 Ctrl+F7）进行编译，编译成功后（见图 1.8）生成目标文件 program1.obj，再单击工具栏中  按钮（或按快捷键 F7）进行连接，连接成功后（见图 1.9）生成可执行文件 project1.exe。

7. 执行应用程序

单击  按钮（或按快捷键 Ctrl+F5）即执行应用程序，弹出运行的 DOS 窗口，显示出运行结果“Welcome to C world!”，其下方的“Press any key to continue”是系统自动加上去的，提示用户可以按任意键退出 DOS 窗口，返回到 Visual C++ 6.0 编辑窗口。运行结果如图 1.10 所示。

0 个错误，编译成功 ←

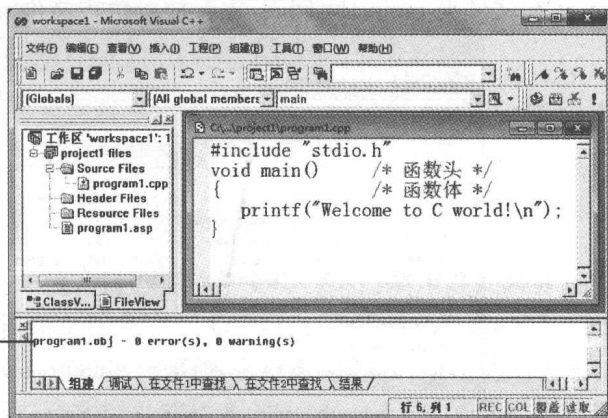


图 1.8 编译成功

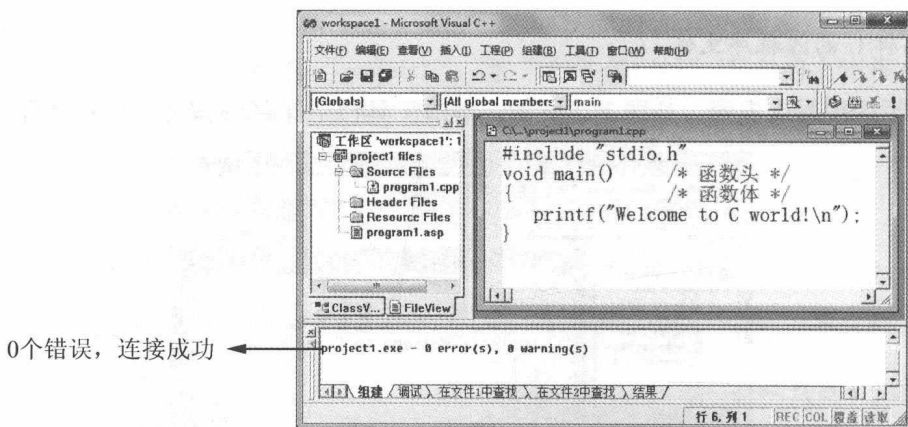


图 1.9 连接成功

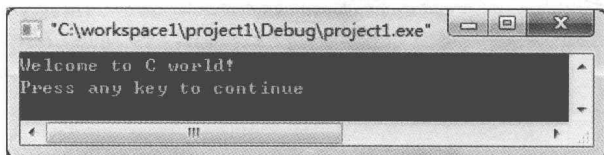


图 1.10 查看运行结果

8. 关闭工作区

完成对 C 源程序的调试后, 为保护好已建立的应用程序, 应执行【文件】→【关闭工作空间】命令, 正确关闭工作区。

9. 查看 C 源程序文件、可执行文件的存放位置

经过编辑、编译、连接和执行 4 个环节后, 在工程文件夹 project1 和 project1\debug 中会产生一些相关的文件。在工程文件夹 project1 中存放源程序文件 program1.cpp, 如图 1.11 所示。在文件夹 project1\debug 中存放可执行文件 project1.exe, 如图 1.12 所示。

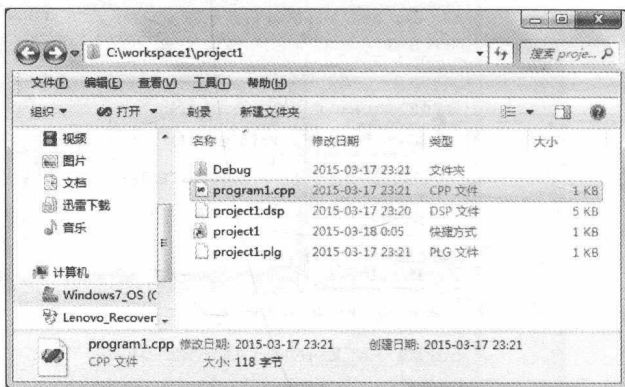


图 1.11 查看源程序文件

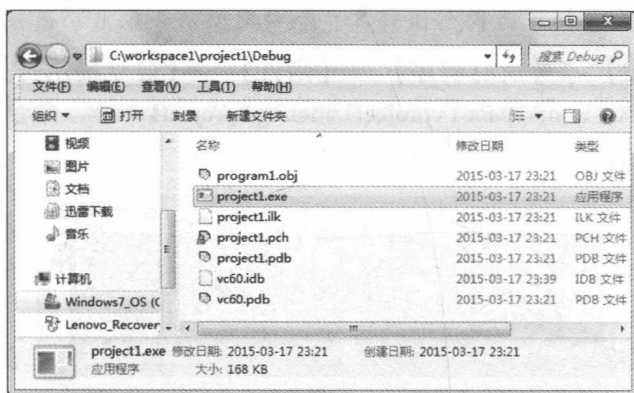


图 1.12 查看可执行文件

10. 再次打开文件

如果要再次打开 C 源程序文件，可以使用以下两种方法。

(1) 选择【文件】→【打开】命令(或按快捷键 Ctrl+O)打开如图 1.13 所示对话框。选中源程序文件 program1.cpp，然后单击【打开】按钮，即可打开该源程序文件，进行重新编辑、编译、连接和运行。

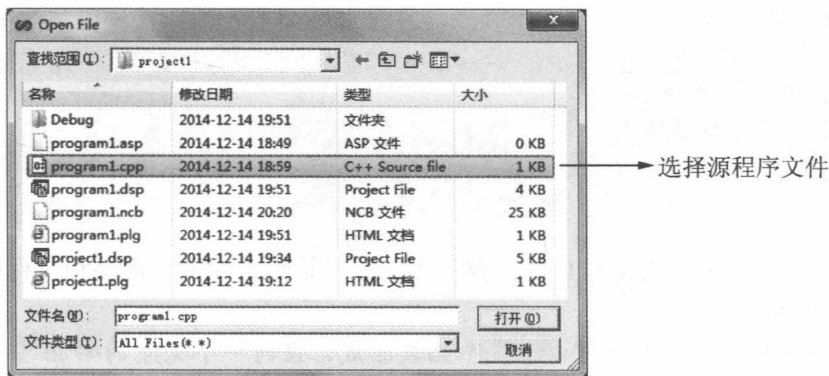


图 1.13 打开源程序文件

(2) 在“资源管理器”中找到文件夹“C:\workspace1\project1”，如图 1.11 所示。双击源程序文件 program1.cpp，也可以打开源程序文件。

11. 运行可执行文件

如果要脱离 Visual C++ 6.0 环境，运行可执行文件，可以使用以下两种方法。

(1) 在“资源管理器”中找到文件夹“C:\workspace1\project1\debug”，如图 1.12 所示。双击可执行文件 project1.exe，可以运行可执行文件。但这时弹出的结果窗口可能会闪退。解决方法：在源程序中最后一个花括号前加个“getchar(;)”函数语句，然后重新编译、连接生成可执行文件。再在图 1.12 所示的资源管理器中双击可执行文件 project1.exe，