



普通高等教育“十三五”规划教材 计算机系列

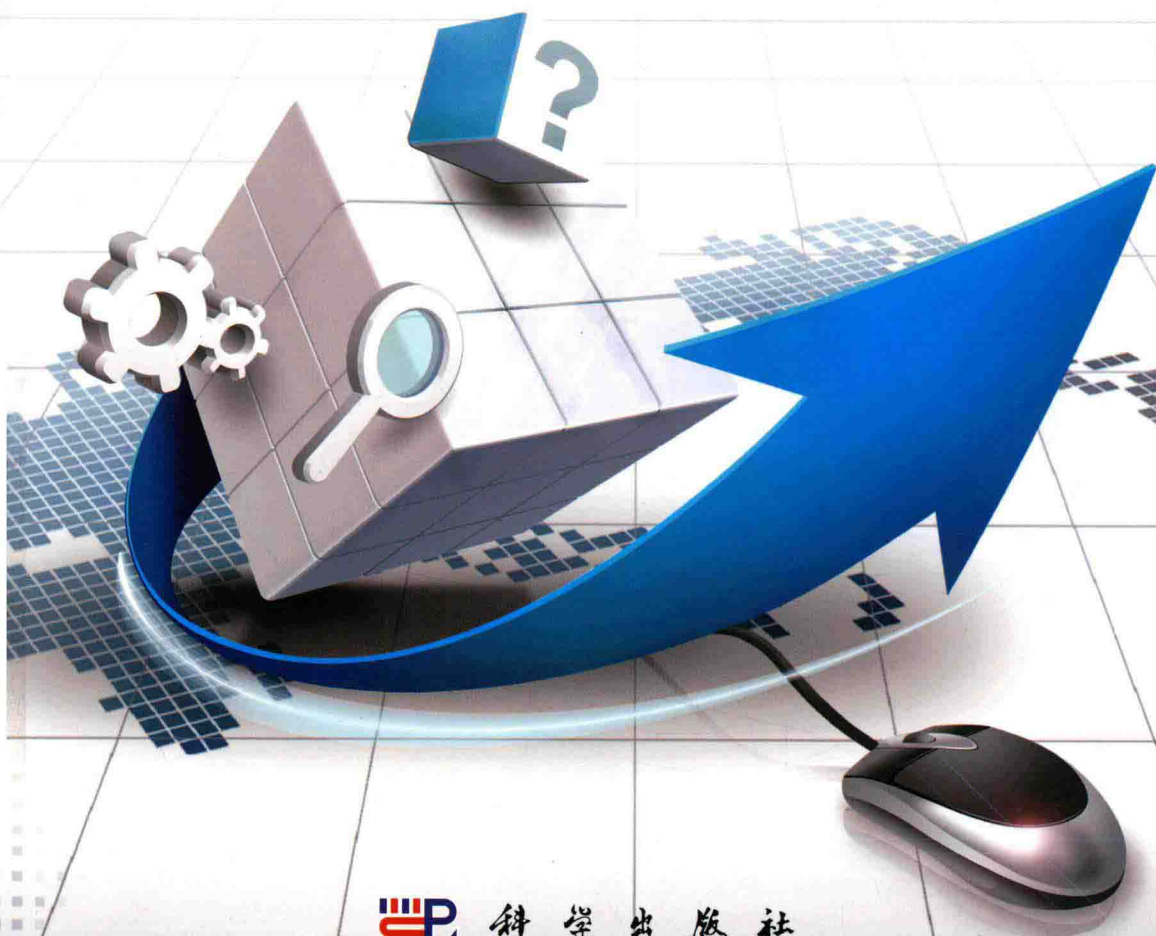
101010

10101001010111010101110

10101010

C语言程序设计 实训教程

蔡春华 司巧梅 樊持杰 主 编
许艳惠 肖 湘



科学出版社

普通高等教育“十三五”规划教材 计算机系列

C 语言程序设计实训教程

蔡春华 司巧梅 樊持杰 主 编
许艳惠 肖 湘
王延松 副主编

本书全面覆盖及实训中涉及的程序均在 Visual C++ 6.0 环境下调试通过。

本书由蔡春华、司巧梅、樊持杰、许艳惠、肖湘、王延松等编写。

定价：00.80 元

（本书为“十二五”普通高等教育本科国家级规划教材）

科学出版社

ISBN 978-7-03-041130-1

内 容 简 介

本书是《C 语言程序设计案例教程》(孙强、宋丽、刘志宇、宋阳、磨然主编,科学出版社)的配套实训教材。本书共分为 12 章,每一章又包括学习目标、知识要点、典型例题、实验指导。学习目标、知识要点、典型例题帮助读者更好地掌握和理解各章的知识点、难点与易错点;实验指导中给出了在 Visual C++ 6.0 集成环境下编辑、编译、调试和运行程序的方法,以及精心设计的具有代表性的实验及相关的实验操作提示。为便于学生检验学习效果,本书提供综合练习及其参考答案。

本书既适于高等院校计算机相关专业师生使用,又可作为自学者的参考用书和计算机培训的辅助教材。

图书在版编目(CIP)数据

C 语言程序设计实训教程/蔡春华等主编. —北京:科学出版社,2017
(普通高等教育“十三五”规划教材 计算机系列)

ISBN 978-7-03-051217-8

I. ①C… II. ①蔡… III. ①C 语言-程序设计-高等学校-教材
IV. ①TP312.8

中国版本图书馆 CIP 数据核字 (2016) 第 321307 号

责任编辑:戴 薇 副 杨 / 责任校对:张 曼
责任印制:吕春珉 / 封面设计:陈方人华平面设计部

科学出版社出版

北京东黄城根北街 16 号

邮政编码:100717

<http://www.sciencep.com>

三河市良远印务有限公司印刷
科学出版社发行 各地新华书店经销

*

2017 年 7 月第 一 版 开本:787×1092 1/16

2017 年 12 月第二次印刷 印张:16

字数:368 000

定价:38.00 元

(如有印装质量问题,我社负责调换〈良远印务〉)

销售部电话 010-62136230 编辑部电话 010-62135927-2032

版权所有,侵权必究

举报电话:010-64030229; 010-64034315; 13501151303

前 言

“C 语言程序设计”是一门理论和实践相结合的课程,学生不仅需要掌握 C 语言程序设计的理论知识,还必须经过大量的实践训练,以培养其程序设计的思维能力和通过编程解决相关专业领域问题的能力。本书是配合《C 语言程序设计案例教程》编写的实训教材,以“提高学生的实践能力,培养学生的编程能力”为宗旨,力求更好地达到教学目的,提高学习者的学习兴趣,培养读者的发散性思维、批判性思维和创造性思维,结合一线教师的多年教学实践经验编写而成。在编排程序设计的内容顺序方面,本书与《C 语言程序设计案例教程》的课程体系相吻合,做到循环渐进,系统学习,广泛实践,便于学生接受。

本书全部习题及实验中涉及的程序都已在 Visual C++ 6.0 环境下调试通过。

本书由牡丹江师范学院蔡春华、司巧梅、樊持杰、肖湘和吉林工程技术师范学院许艳惠担任主编,由吉林工程技术师范学院王延松担任副主编。本书具体编写分工如下:第 1~3 章由司巧梅编写,第 4~7 章由蔡春华和许艳惠共同编写,第 8~10 章由樊持杰和王延松共同编写,第 11 章由司巧梅和樊持杰共同编写,第 12 章由肖湘编写。其中,第 10~12 章为选学内容。全书由蔡春华统稿。

本书得到黑龙江省教育厅备案项目(项目编号为 1351MSYYB014)、牡丹江市科技计划项目(项目编号为 Z2016s0030)、牡丹江师范学院教学改革项目(项目编号为 16-JG18047)、牡丹江师范学院科研项目(项目编号为 YB201601)及牡丹江师范学院教学改革项目(项目编号为 16-JG18046)的支持。

在组织和编写本书过程中,编者借鉴了同类图书和互联网上的相关科研、教学资料。限于篇幅,这些参考资料未在本书中一一列出,在此对这些作者深表谢意。

由于编者水平有限,加之时间仓促,书中难免存在不妥和疏漏之处,恳请广大读者批评指正,以便在今后再版时对本书进行完善。

编 者

2016 年 11 月

目 录

第 1 章 C 语言概述	1
1.1 知识要点	1
1.1.1 程序设计语言及其发展	1
1.1.2 C 语言程序的基本结构	2
1.1.3 C 语言程序的运行	2
1.2 典型例题	3
1.3 实验指导	7
第 2 章 C 语言程序设计基础知识	18
2.1 知识要点	18
2.1.1 标识符、常量和变量	18
2.1.2 数据的类型	19
2.1.3 数据类型的转换	22
2.1.4 运算符和表达式	23
2.2 典型例题	28
2.3 实验指导	29
第 3 章 顺序结构程序设计	35
3.1 知识要点	35
3.1.1 C 语言语句	35
3.1.2 数据输出、输入函数	36
3.1.3 顺序结构程序设计	39
3.2 典型例题	39
3.3 实验指导	41
第 4 章 选择结构程序设计	50
4.1 知识要点	50
4.1.1 用 if 语句构成单分支选择结构	50
4.1.2 用 switch 语句构成多分支选择结构	53
4.2 典型例题	54
4.3 实验指导	56



第 5 章 循环结构程序设计	67
5.1 知识要点	67
5.1.1 while 循环结构	67
5.1.2 do...while 循环结构	68
5.1.3 for 循环结构	69
5.1.4 循环嵌套	71
5.1.5 break 语句和 continue 语句的作用	72
5.2 典型例题	73
5.3 实验指导	76
第 6 章 编译预处理	84
6.1 知识要点	84
6.1.1 宏定义	84
6.1.2 文件包含	86
6.1.3 条件编译	87
6.2 典型例题	88
6.3 实验指导	89
第 7 章 函数	95
7.1 知识要点	95
7.1.1 函数概述	95
7.1.2 函数的定义	95
7.1.3 函数调用	97
7.1.4 函数说明	97
7.1.5 参数传递	98
7.1.6 函数的嵌套调用和递归调用	98
7.1.7 变量的作用域	99
7.1.8 变量的存储类别	100
7.2 典型例题	102
7.3 实验指导	104
第 8 章 数组	113
8.1 知识要点	113
8.1.1 一维数组的定义及应用	113
8.1.2 二维数组的定义及应用	114
8.1.3 字符串与字符数组	115

8.1.4 二维字符数组	116
8.2 典型例题	117
8.3 实验指导	118
第9章 指针	125
9.1 知识要点	125
9.2 典型例题	127
9.3 实验指导	129
*第10章 结构体及共用体	140
10.1 知识要点	140
10.1.1 结构体	140
10.1.2 共用体	144
10.2 典型例题	145
10.3 实验指导	146
*第11章 文件管理	155
11.1 知识要点	155
11.1.1 概述	155
11.1.2 文件的打开与关闭	156
11.1.3 文件的读写	159
11.1.4 文件的定位	162
11.2 典型例题	163
11.3 实验指导	167
*第12章 C语言高级程序设计	173
12.1 知识要点	173
12.1.1 位运算	173
12.1.2 动态存储分配	176
12.1.3 链表	178
12.2 典型例题	189
12.3 实验指导	199
综合练习	207
综合练习1	207
综合练习2	214
综合练习3	221



综合练习答案	229
附录 1 C 语言编译错误信息汇编	234
附录 2 C 语言中的关键字	241
附录 3 运算符与结合性	242
附录 4 ASCII 码字符表	243
附录 5 C 语言常用库函数	244
参考文献	248

第1章 C语言概述

学习目标

- (1) 了解程序设计语言及其发展。
- (2) 掌握C语言的程序结构。
- (3) 掌握C语言的运行环境。
- (4) 掌握C语言的运行方法。
- (5) 领会C语言程序设计的风格。

1.1 知识要点

1.1.1 程序设计语言及其发展

用于书写计算机程序的语言称为程序设计语言，它是人与计算机之间进行信息交流的工具。

程序设计语言的种类非常多，总的来说，可以分为机器语言、汇编语言和高级语言3大类。

1. 机器语言

机器指令的集合构成了机器语言，用机器语言编写的程序就是机器语言程序。计算机所能识别的语言只有机器语言，但机器语言非常难以记忆和识别。通常，人们编程时不采用机器语言，而采用汇编语言和高级语言。

2. 汇编语言

汇编指令、伪指令、宏指令和汇编程序一起组成了汇编语言。汇编语言直接面向机器，用汇编语言编写程序简洁、快速，常用于对运行速度要求较高的实时控制等场合。用汇编语言编制的用户程序称为汇编语言源程序。汇编语言的实质和机器语言是相同的，都是直接对硬件操作，但指令采用了英文缩写的标识符，更容易识别和记忆。而其所占用的存储空间和执行速度与机器语言相仿。

3. 高级语言

高级语言主要是相对于汇编语言而言的，它是以较接近自然语言和数学公式的形式编程，基本脱离了机器的硬件系统，用人们更易于理解的方式编写程序。高级语言并不



特指某一种具体的语言，而包括了很多编程语言，如 BASIC 语言、C 语言、FoxPro 等，这些语言的语法和命令格式各不相同。

高级语言所编制的程序不能直接被计算机识别，必须经过转换才能被执行，按照转换方式可将高级语言分为两类。

1) 解释类

其执行方式类似于日常生活中的“同声翻译”，应用程序源代码一边由相应语言的解释器“翻译”成目标代码（机器语言），一边执行，因此效率比较低，不能生成可独立执行的可执行文件，应用程序不能脱离其解释器。但这种方式比较灵活，可以动态地调整、修改应用程序，如 BASIC 语言便采用了这种方式。

2) 编译类

编译是指在应用源程序执行之前，将程序源代码一次性“翻译”成目标代码（机器语言），然后在机器上运行目标程序，因此其目标程序可以脱离语言环境独立执行，使用比较方便且效率较高。但应用程序一旦需要修改，就必须先修改源代码，再重新编译生成新的目标文件才能执行。如果只有目标文件而没有源代码，修改会很不方便。现在大多数的编程语言是编译型的，如 C 语言等。

1.1.2 C 语言程序的基本结构

(1) C 语言程序是由函数组成的。一个完整的 C 语言程序可以由一个或多个函数组成，其中 main 函数（主函数）必不可少，且只有一个。C 语言程序执行时，总是从 main 函数开始，与 main 函数在整个程序中的位置无关，其他函数都是为 main 函数服务的。函数是 C 语言程序的基本单位，用函数来实现特定的功能，所以 C 语言是函数式语言。C 语言的函数包括系统提供的库函数（如 printf 函数），以及用户根据实际问题编制设计的函数。

(2) 源程序中可以有预处理命令，预处理命令通常放在源文件或源程序的最前面。

(3) 每一条语句都必须以分号结尾，但预处理命令、函数头和右大括号“}”之后不加分号。

(4) 注释不是程序的必需部分，在程序执行时注释不起任何作用。注释的作用是增加程序的可读性，因此，适当地在程序中加入注释，是一种良好的程序设计风格。C 语言有块注释和行注释两种注释方法。

(5) 在 C 语言中，虽然一行可写多条语句，一条语句也可占多行，但建议一行只写一条语句。

(6) 一般采用缩进格式书写程序，以提高程序的可读性和清晰性。

1.1.3 C 语言程序的运行

1. 运行 C 语言程序的步骤

C 语言属于编译型编程语言，要使 C 语言程序在计算机上执行，必须经过源程序的

编辑、编译和连接等一系列步骤，最后得到可执行程序并运行。

1) 编辑

编辑是建立或修改 C 语言源程序文件的过程，并以文件的形式存储在磁盘上，C 语言源程序文件的扩展名为.c。

2) 编译

C 语言编译器将 C 语言源程序转换为机器代码，生成目标程序，目标程序文件的扩展名为.obj。在 C 语言源程序的编译过程中，可以检查出程序中的语法错误。

3) 连接

机器不能直接执行编译生成的目标程序，还需将目标程序与库文件进行连接处理，连接工作由连接程序完成。经过连接后，生成可执行程序，可执行程序的扩展名为.exe。

4) 运行

C 源程序经过编译、连接后生成了可执行文件。生成的可执行文件既可在编译系统环境下运行，又可以脱离编译系统直接在操作系统中执行。

当编译出现错误时，说明 C 语言程序中有语法错误；若在运行时出现错误或结果不正确，则说明程序设计有错误（称为逻辑错误），需要修改源程序并重新编译、连接和运行，直至将程序调试正确为止。

2. 集成开发环境

集成开发环境提供编程时所必需的工具，这些工具有编辑器、编译器和调试器，它们集成在一个软件包内供程序员使用。在早前的 DOS 环境下主要的 C 语言集成开发环境有 TC 2.0、Turbo C++ 3.0 和 Borland C++。在 Windows 环境下主要使用 Visual C++ 6.0，Visual C++ 6.0 既可以对 C++ 进行编译，也可以对 C 进行编译。本书的示例程序均可在 Visual C++ 6.0 集成环境下进行编辑、编译和运行。

1.2 典型例题

【例 1-1】编写问候程序，输出字符串“hello,world”。

源程序：

```
#include<stdio.h>
main()
{
    printf("hello, world\n");
}
```

程序分析：一个 C 语言程序，无论规模大小，都是由函数组成的，函数中包含一些语句，以指定所要执行的操作，本例中函数的名称为 main。通常情况下，C 语言并没有限制函数必须取一个什么样的名称，但 main 是特殊的函数，称为主函数，每个程序都



以 main 函数为起点开始执行,这就意味着每个程序都必须包含一个 main 函数,函数体必须由 { } 括起来。

C 语言编译系统将一些常用的操作或计算功能定义成函数,如 printf、scanf、sqrt、fabs 等,这些函数称为标准库函数,其说明部分放在指定的以.h 为扩展名的头文件中,如存放标准输入/输出库函数说明的头文件名为 stdio.h,在使用系统库函数时必须将对应的头文件包含进来。#include <stdio.h>是一个预处理命令,以#开始,其功能是包含文件“stdio.h”。

在 printf("hello, world\n");语句中,printf 函数是一个用于输出的库函数,在本例中被主函数 main()调用,它用于输出双引号中的字符串。双引号中的字符序列称为字符串或字符串常量,"hello,world\n"就是一个字符串,是 printf 函数的参数,"hello, world"是原样输出的字符串序列,printf 函数不会自动换行,\n 是换行符,遇到它时输出将换行。

printf("hello, world\n");语句最后以“;”表示该语句结束。

【例 1-2】求两个整数 10 和 20 的和并输出结果。

源程序:

```
#include<stdio.h> /*包含头文件 stdio.h*/
main()
{
    int a, b, sum; /*定义变量*/
    a=10; /*给变量 a 赋整数值 10*/
    b=20; /*给变量 b 赋整数值 20*/
    sum=a+b; /*求和*/
    printf("sum=%d\n", sum); /*输出 sum 的值*/
}
```

运行结果:

```
sum=30
```

程序分析:在本程序中,/*...*/是注释,不是程序的必需部分,在程序执行时注释不起任何作用。注释的作用是增加程序的可读性,因此,适当地在程序中加入注释,是一种良好的程序设计风格。

C 语言的注释方法有以下两种。

(1) 块注释。形式:

```
/*
```

```
...
```

```
注释内容
```

```
...
```

```
*/
```

这是跨多行的注释语句,适用于注释多行,“/*”和“*/”之间的内容为注释内容。

(2) 行注释。形式:

```
/* ...注释内容... */
```

行注释放在一行上,通常放在语句之后。

注释可以出现在程序中的任何位置,注释中的任何内容都不会被计算机执行。

程序中 `int a, b, sum;` 语句定义 `a`、`b`、`sum` 为 `int` 类型变量,在 C 语言和其他大部分编程语言中,使用变量 (Variables) 来存储数据。每个变量都由一个变量名来标识。每个变量必须有一种类型,用于表示它所存储的是哪种类型的数据。C 语言的基本数据类型有整型、实型、字符型等。变量必须先定义,然后才能使用。

变量定义的一般形式:

类型标识符 变量名列表

变量定义后其初值一般是不确定的,不能直接使用。例如:

```
int a; printf("%d\n", a);          /******错误!*****/  
int a; a=10; printf("%d\n", a);    /******正确*****/
```

在定义变量的同时也可以对变量赋初值,称为变量的初始化。

例如,语句 `int sum=0;` 定义了 `sum` 为 `int` 类型变量,为 `sum` 赋初值 0。

又如, `printf("sum=%d\n", sum);` 语句用于输出 `sum` 的值, C 语言中的 `printf` 是使用很普遍的命令,称为格式输出函数。其基本命令格式如下:

```
printf(格式控制, 输出列表);
```

其中,格式控制部分用双引号引起来,里面通常包含两种信息:一种是普通字符,普通字符按原样输出;另一种是以“%”开头的格式说明,其作用是将数据按指定的格式输出。例如,“%d”表示对应的输出值(即 `sum` 的值)以十进制整数形式显示,其余都是普通字符。所以,此程序执行后输出的结果如下:

```
sum=30
```

对应关系如图 1-1 所示。

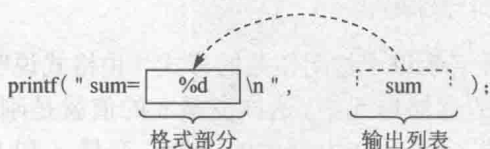


图 1-1 printf 函数输出示例 (一)

如果在程序执行后想得到如下形式的输出结果:

```
10+20=30
```

则程序中的 printf 语句可改写为

```
printf("%d+%d=%d\n", 10, 20, sum);
```

其中, 10、20、sum 为输出列表, 各表项之间用逗号分开。由于有 3 个输出值, 所以用 3 个格式说明符与之——对应, 如图 1-2 所示。

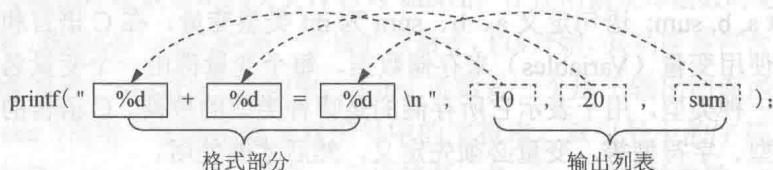


图 1-2 printf 函数输出示例 (二)

该程序只能计算 10 与 20 的和, 因为程序中规定了 a 和 b 的值, 如果计算其他两个数的和, 则需要修改程序。程序在运行时通过键盘输入需要求和的两个数, 然后进行求和计算并输出, 这种实现方式最为理想。

【例 1-3】求任意两个整数的和并输出结果。

源程序:

```
#include<stdio.h>
main()
{
    int a, b, sum;          /*定义变量*/
    scanf("%d", &a);        /*输入第一个整数*/
    scanf("%d", &b);        /*输入第二个整数*/
    sum=a+b;                /*计算和*/
    printf("The sum of %d and %d is %d\n", a,b,sum); /*输出和*/
}
```

程序分析: 在 C 语言中, 要想获得通过键盘输入的值, 可以使用 scanf 函数, scanf 函数是与 printf 函数相对应的格式输入函数, 其基本命令格式如下。

scanf (格式控制, 地址列表);

语句 scanf("%d", &a); 表示以十进制整数的形式 (由格式说明符 "%d" 指定) 输入数据, 存放到变量 a 对应的存储单元中, 这样变量 a 的值就是刚刚通过键盘输入的值, & 是取地址运算符, &a 指变量 a 在内存中的地址。变量 a 和 b 及其所存储的数值如图 1-3 所示。



图 1-3 变量及其值

1.3 实验指导

◆ 实验目的

- (1) 了解 Visual C++ 6.0 (以下简称 VC 6.0) 集成开发环境。
- (2) 掌握 C 语言程序的基本结构。
- (3) 学会如何在 VC 6.0 集成开发环境中编辑、编译、连接和运行一个 C 语言程序。

◆ 实验内容

【实验 1】在 VC 6.0 集成开发环境中编辑、编译、连接和运行一个 C 语言程序。

VC 6.0 由 Microsoft 公司开发,它不仅是一个 C++编译器,而且是一个基于 Windows 操作系统的可视化集成开发环境 (Integrated Development Environment, IDE)。VC 6.0 由许多组件组成,包括编辑器、调试器,以及程序向导 AppWizard、类向导 Class Wizard 等开发工具。自 1993 年 Microsoft 公司推出 Visual C++ 1.0 后,随着其新版本的不断问世,Visual C++已成为专业程序员进行软件开发的首选工具。虽然 Microsoft 公司推出了 Visual C++.NET,但它的应用有很大的局限性,只适用于 Windows 2000、Windows XP 和 Windows NT 4.0。所以,实际中大多以 VC 6.0 为平台。

由于 C++是由 C 语言发展起来的,因此它支持 C 语言的编译。VC 6.0 是使用最多的版本,其最大的缺点是对模板的支持比较差。现在最新补丁程序为 SP6,推荐安装,否则编译时易出现假死状态。VC 6.0 仅支持 Windows 操作系统。

(1) 通过选择“开始”|“程序”|“Microsoft Visual Studio 6.0”|“Microsoft Visual C++ 6.0”命令,或者双击桌面上的 VC 6.0 快捷方式图标即可启动 VC 6.0,启动后的界面如图 1-4 所示。

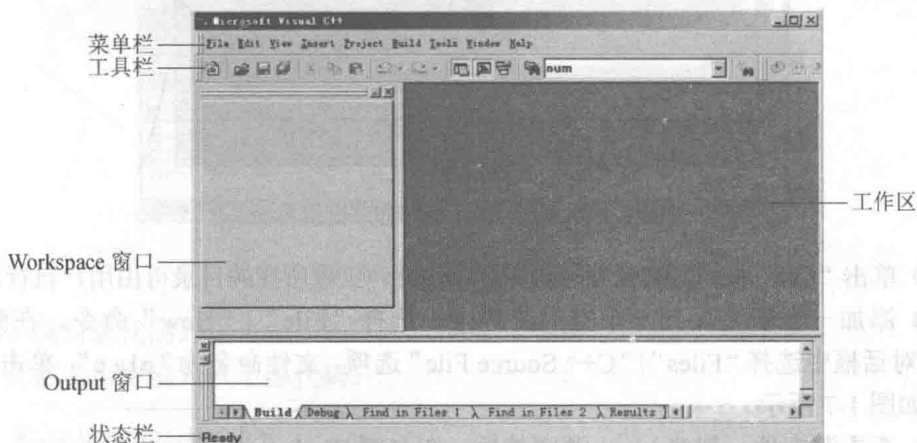


图 1-4 VC 6.0 启动后的界面



(2) 选择“File”|“New”命令，弹出如图 1-5 所示的对话框。选择“Win32 Console Application”选项，并在右侧的“Project name”文本框中输入工程名“lab01”，在“Location”文本框中指定保存目录，单击“OK”按钮。

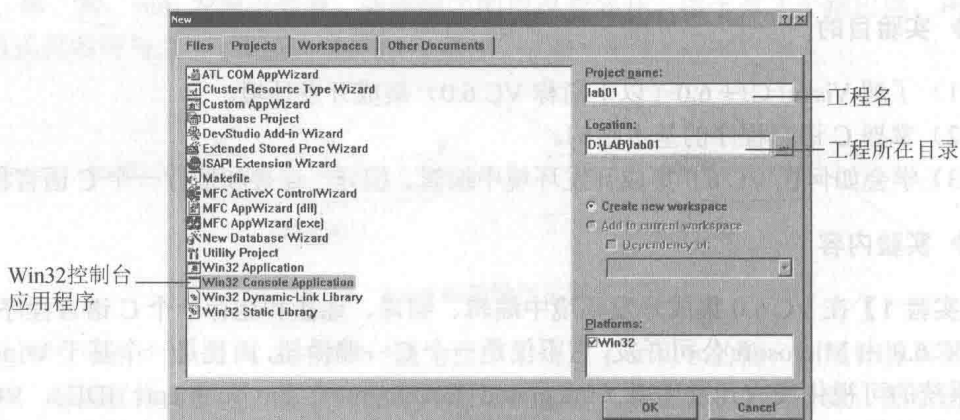


图 1-5 创建工程

选中“An empty project.”单选按钮，即默认选项，单击“Finish”按钮，如图 1-6 所示。

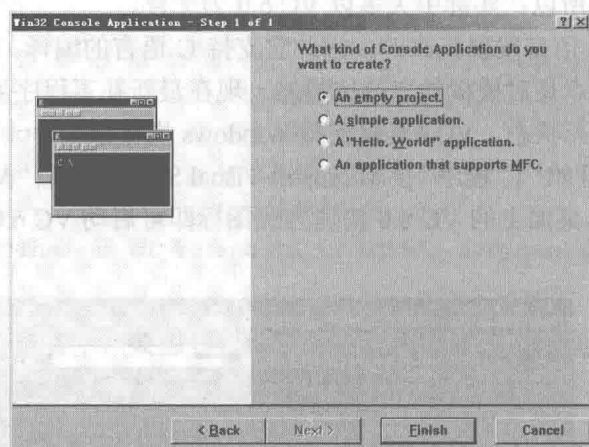


图 1-6 Win32 控制台应用程序配置界面

(3) 单击“OK”按钮，确认创建新工程（注意：工程所在的目录可由用户自行创建）。

(4) 添加一个新文件到一个空的工程中。选择“File”|“New”命令，在弹出的“New”对话框中选择“Files”|“C++ Source File”选项，文件命名为“pla.c”，单击“OK”按钮，如图 1-7 所示。

(5) 查看源文件：创建 pla.c 源文件后，在左侧 Workspace 窗口中的 FileView 标签下，从 Source Files 结点中可以看到该 pla.c 文件，如图 1-8 所示。

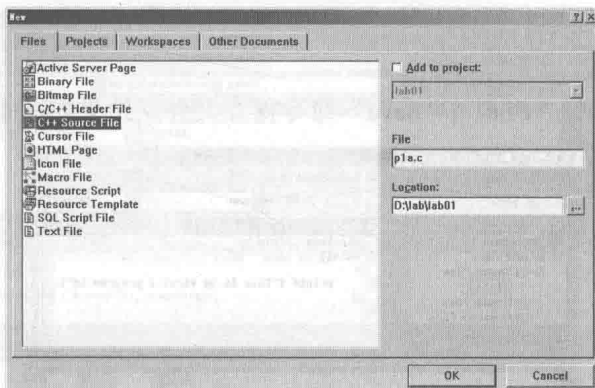


图 1-7 添加源文件

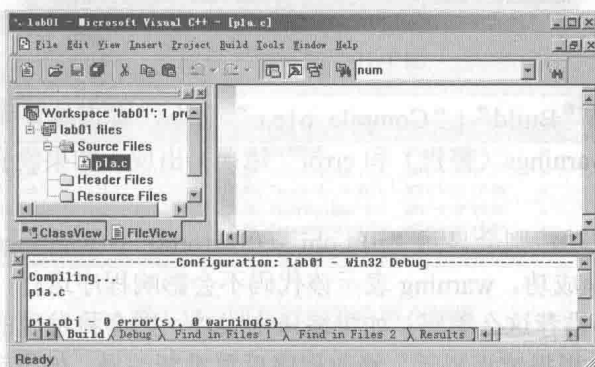


图 1-8 文档窗口

在 Windows 资源管理器中，也可以发现该文件被物理地创建了，如图 1-9 所示。



图 1-9 Windows 资源管理器中的 pla.c 文件

(6) 编写源代码。

在文档窗口中输入以下源代码：

```
#include<stdio.h>
main()
{
```