

高等学校计算机专业规划教材

C语言程序设计案例实践 (第2版)



李丹程 刘莹 那俊 编著

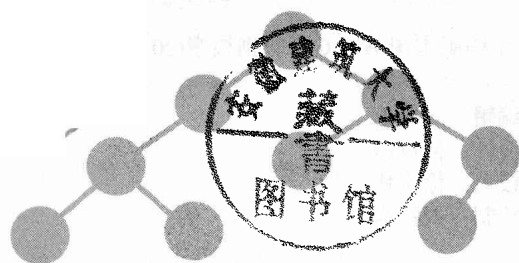
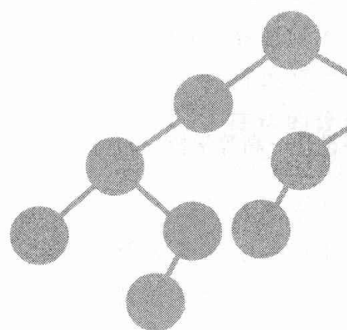


清华大学出版社

高等学校计算机专业规划教材

C语言程序设计案例实践 (第2版)

李丹程 刘莹 那俊 编著



清华大学出版社
北京

内 容 简 介

本书涵盖了 C 语言的重要基础知识, 重点介绍 C 语言的综合运用, 通过详细实例, 结合具体的实际应用需求, 循序渐进地启发学生独立完成实践题目, 培养学生实际分析问题、解决问题的能力。本书设计了 10 个信息管理实例项目, 并介绍了两种 C 语言程序开发环境的使用方法, 以满足不同学校和学生的程序实践要求。

本书不仅可以作为计算机及相关专业的 C 语言实践课程教材, 而且也可以作为其他 C 语言教材的参考书, 亦可作为高等学校各专业 C 语言教辅教材。

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

C 语言程序设计案例实践 / 李丹程, 刘莹, 那俊编著. —2 版. —北京: 清华大学出版社, 2018
(高等学校计算机专业规划教材)

ISBN 978-7-302-48687-9

I. ①C… II. ①李… ②刘… ③那… 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(2017)第 271396 号

责任编辑: 龙启铭

封面设计: 何凤霞

责任校对: 徐俊伟

责任印制: 刘祎淼

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社 总 机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者: 三河市少明印务有限公司

经 销: 全国新华书店

开 本: 185mm×260mm

印 张: 12

字 数: 263 千字

版 次: 2009 年 2 月第 1 版

2018 年 1 月第 2 版

印 次: 2018 年 1 月第 1 次印刷

印 数: 1~2000

定 价: 29.00 元

产品编号: 075837-01



程序设计实践是对编程能力的综合训练,对培养具有创新意识、创新能力的高素质软件人才有很大的推动作用。通过采用项目实例来进行 C 语言编程的综合训练,可以提高学生实际分析问题、解决问题、编程实践、自主创新的能力,同时,还可以培养学生的团队合作精神、创新协作意识及勇于探索的科学精神。本书提供的程序实践题目是基于日常生活中常见的应用领域,可以提高学生的编程乐趣,切身体会 C 语言编程在现实生活中发挥的作用。

本程序实践的特点如下:

(1) 涵盖了 C 语言的重要基础知识,包括三种基本程序结构、结构体、链表、文件的使用等,重点放在结构体的处理方面。

(2) 每章的设计要求均结合实际应用的需求,从而激发学生的兴趣,调动学生主动学习的积极性,引导学生按照实际需要进行编程实践。

(3) 采用引导式学习的方式,通过详细实例介绍 Visual Studio 6.0 开发环境及 CodeBlocks 开发环境下的工程建立、调试、测试方法,将设计要求、算法和源程序分开,便于学生进行独立思考,学生可以在理解要求的基础上,脱离书中提供的代码来完成任务。

(4) 提供思考题来引导学生拓展自己的思维,对实践过程进行总结和归纳。

(5) 全书的代码按照规范编写,给学生提供良好的范例,培养学生严谨的编程态度和良好的编程风格。

全书共分 11 章,第 0 章预备知识部分介绍本程序实践使用的开发环境;第 1~10 章提供了学生熟悉的 10 个生活中常用的信息管理项目,锻炼学生解决实际问题的能力。

参加本书编写的还有柯章瀚、石凯、王学宾、滕志扬、廖丁石、于鹤鑫、闫振南、王重、刘景明、艾萌、郑晨、贾若、袁铮、黄守凯,他们参与了部分章节的编写、程序调试和校对等工作,为本书的出版付出了大量的精力。此外,朱志良教授对本书出版给予了大力支持。在此向他们表示衷心的感谢。

由于时间仓促及编者水平有限,书中不足之处在所难免,敬请读者批评指正。



第 0 章 预备知识：开发环境 / 1

0.1	Visual C++ 6.0 开发环境	1
0.1.1	Visual C++ 6.0 的安装和启动	1
0.1.2	编辑一个 C 源程序	1
0.1.3	编译、链接和运行 C 程序	6
0.1.4	C 程序的调试	8
0.2	CodeBlocks 开发环境	10
0.2.1	CodeBlocks 的安装和启动	10
0.2.2	编辑一个 C 源程序	11
0.2.3	编译、链接和运行 C 程序	15
0.2.4	C 程序的调试	16

第 1 章 学生信息管理 / 19

1.1	实践目的	19
1.2	基本要求	19
1.3	算法分析	20
1.4	参考代码	22
1.5	代码测试	53
1.6	思考题	55

第 2 章 旅馆信息管理 / 56

2.1	实践目的	56
2.2	基本要求	56
2.3	算法分析	57
2.4	参考代码	60
2.5	代码测试	76
2.6	思考题	77

第 3 章 餐厅信息管理 / 78

3.1	实践目的	78
-----	------------	----



3.2 基本要求	78
3.3 算法分析	79
3.4 参考代码	81
3.5 代码测试	91
3.6 思考题	92

第 4 章 职工信息管理 / 93

4.1 实践目的	93
4.2 基本要求	93
4.3 算法分析	94
4.4 参考代码	96
4.5 代码测试	111
4.6 思考题	113

第 5 章 设备信息管理 / 114

5.1 实践目的	114
5.2 基本要求	114
5.3 算法分析	115
5.4 参考代码	117
5.5 代码测试	129
5.6 思考题	130

第 6 章 图书信息管理 / 131

6.1 实践目的	131
6.2 基本要求	131
6.3 算法分析	132
6.4 参考代码	134
6.5 代码测试	140
6.6 思考题	141

第 7 章 医院信息管理 / 142

7.1 实践目的	142
7.2 基本要求	142
7.3 算法分析	143
7.4 参考代码	145
7.5 代码测试	154
7.6 思考题	155



第 8 章 超市信息管理 / 156

8.1 实践目的	156
8.2 基本要求	156
8.3 算法分析	157
8.4 参考代码	158
8.5 代码测试	164
8.6 思考题	165

第 9 章 票务信息管理 / 166

9.1 实践目的	166
9.2 基本要求	166
9.3 算法分析	166
9.4 参考代码	170
9.5 代码测试	174
9.6 思考题	175

第 10 章 超级终端仿真 / 176

10.1 实践目的	176
10.2 基本要求	176
10.3 算法分析	176
10.4 参考代码	179
10.5 代码测试	181
10.6 思考题	181

第0章

预备知识：开发环境

本书以 Visual C++ 6.0 或 CodeBlocks 作为 C 源程序的实践开发环境，本章将首先介绍 Visual C++ 6.0 和 CodeBlocks 环境的基本操作，包括 Visual C++ 6.0 和 CodeBlocks 的安装和启动、C 源程序的编辑、运行与调试以及多文件工程的建立和运行方法。

由于 Visual C++ 6.0 程序发布时间较早，适用于 Windows XP 及更早版本的 Windows 系统，对于 Windows 7、Windows 8 系统存在兼容性问题。所以建议使用 Windows 7、Windows 8 及其以上版本系统的读者选择 CodeBlocks 作为进行实践开发的环境。

0.1 Visual C++ 6.0 开发环境

0.1.1 Visual C++ 6.0 的安装和启动

如果计算机上未安装 Visual C++ 6.0，则按照安装向导直接安装，具体步骤此处不再详述。安装过程中建议同时安装 MSDN，以便日后自学。

成功安装 Visual C++ 6.0 后，可以在桌面上看到如图 0.1 所示的图标，或通过“开始”→“程序”→Microsoft Visual Studio 找到 Visual C++ 6.0 启动程序。



图 0.1 Visual C++ 6.0 图标

双击 Visual C++ 6.0 图标，或依次单击“开始”→“程序”→Microsoft Visual Studio→Visual C++ 6.0 选项，即可启动 Visual C++ 6.0 集成开发环境。正常启动开发环境后，可见如图 0.2 所示的主窗口操作界面。

0.1.2 编辑一个 C 源程序

编辑一个 C 源程序的前提是新建一个程序或者打开现有的程序。本节首先介绍如何新建一个程序，如何打开现有程序，并在此基础上对文件进行编辑。

1. 新建一个 C 源程序

(1) 新建一个工程。在 Visual C++ 6.0 主窗口的主菜单栏中，单击 File（文件）菜单项，然后单击 New 菜单项，也可按 Ctrl+N 键进入，如图 0.3 所示。

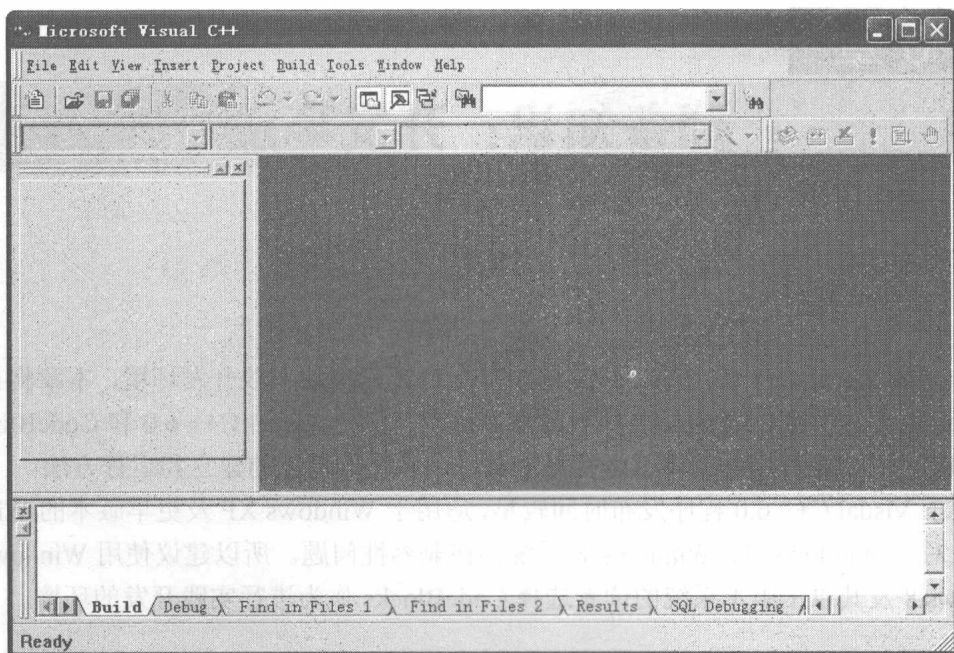


图 0.2 Visual C++ 6.0 主窗口

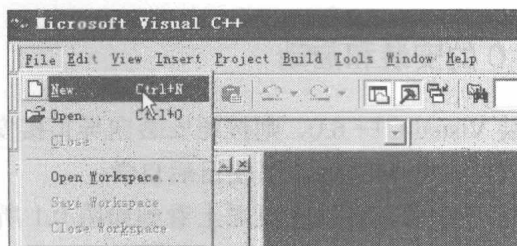


图 0.3 新建工程

屏幕上出现一个 New 对话框, 如图 0.4 所示。单击此对话框上方的 Project 选项卡,

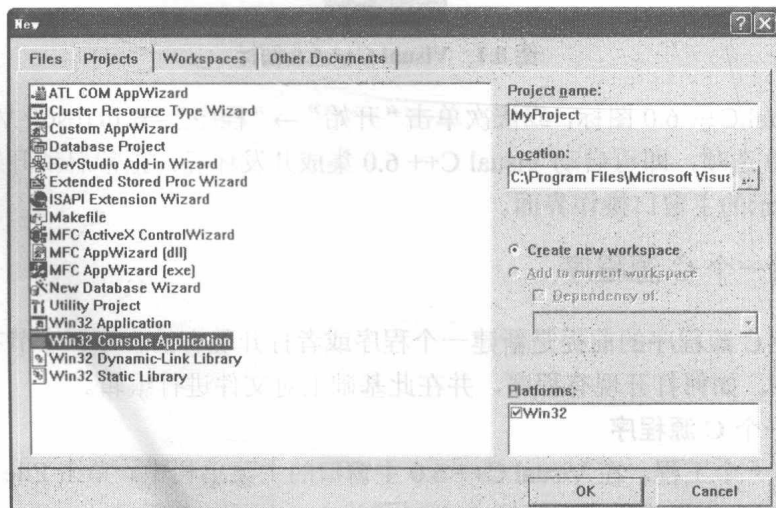


图 0.4 New 对话框

在其下方选择 Win32 Console Application 选项，在 Project name 文本框中输入工程名，在 Location 文本框中设定工程存储路径，然后单击 OK 按钮。

单击 OK 按钮后，会弹出如图 0.5 所示的对话框，选中 An empty project 单选按钮，单击 Finish 按钮。

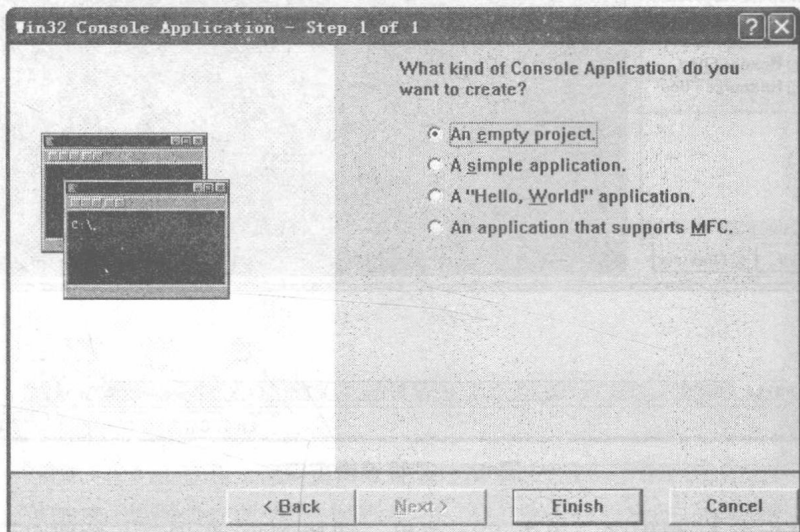


图 0.5 选择工程类型

之后，弹出如图 0.6 所示的对话框。

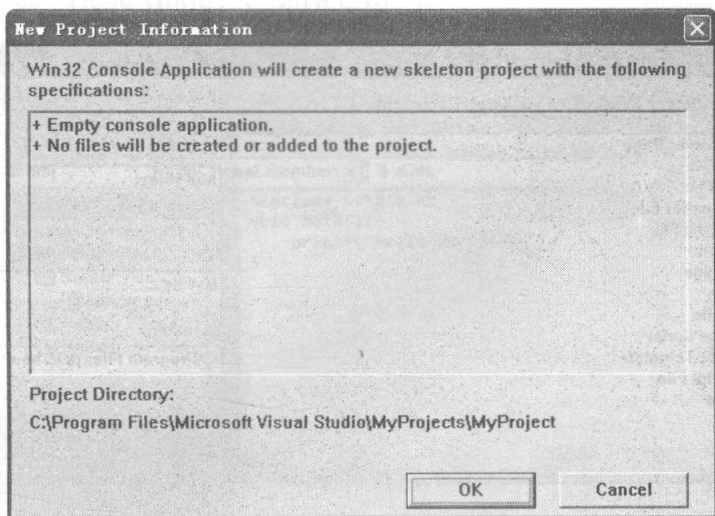


图 0.6 新工程信息

注意：该对话框中显示了工程文件的路径。

单击 OK 按钮，则进入工程编辑窗口，如图 0.7 所示。在工程编辑窗口可以看到 MyProject 工程下只有三个空白的文件夹。

然后，就可以在当前工程下创建一个 C 源程序。

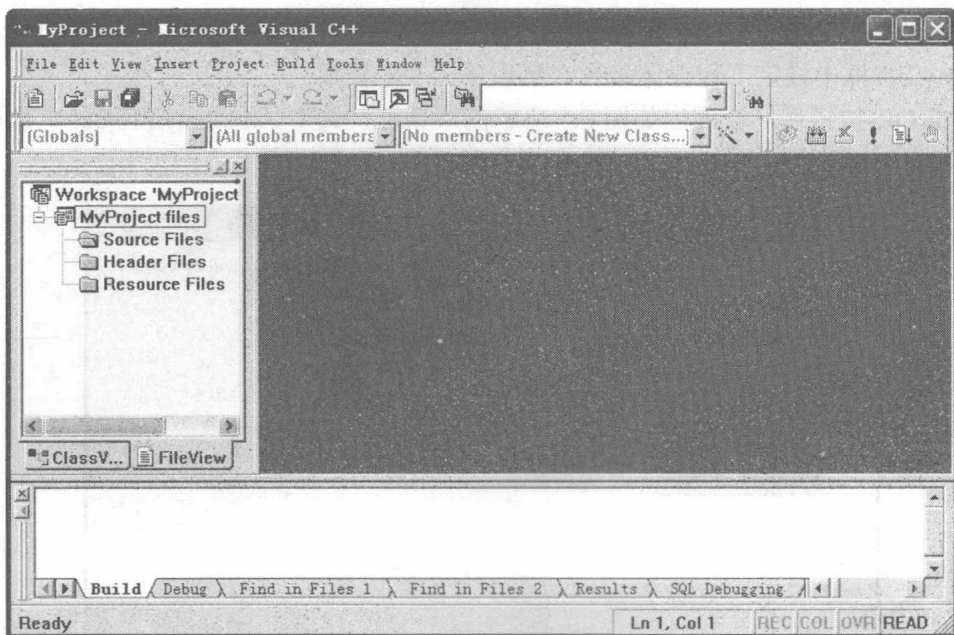


图 0.7 工程编辑窗口

(2) 新建一个 C 源程序。单击 File 菜单, 选择 New 选项。在新建对话框中, 类似新建工程操作 (如图 0.8 所示), 在 Files 选项卡中选择 C++ Source File 选项; 勾选 Add to project, 选择添加到刚新建的工程 MyProject; 在 File 文本框中输入扩展名为.c 的文件名, 单击 OK 按钮。

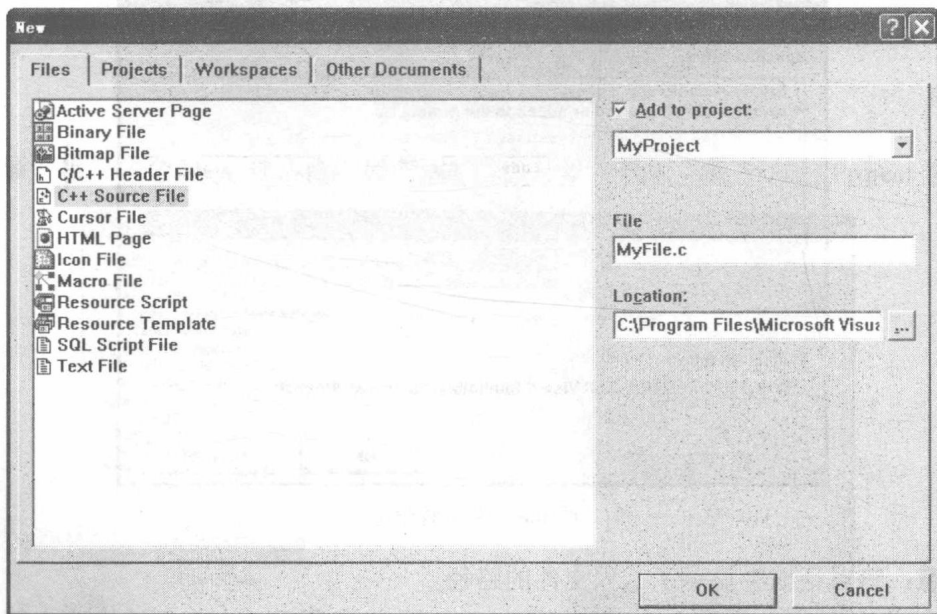


图 0.8 新建 C 源程序

注意: 如果在文件名中不显式地输入扩展名.c, 则 VC++ 将为文件附加上默认扩展名.cpp, 并按照 C++ 语言的语法进行检查。由于 C++ 的语法检查要比 C 语言的语法更为

严格，因此，这里建议显式地输入文件扩展名.c。

单击 OK 按钮后，可以进入 MyFile.c 的编辑窗口，如图 0.9 所示。在该窗口的标题栏和工程窗口中都显示出了当前要编辑的文件名 MyFile.c。

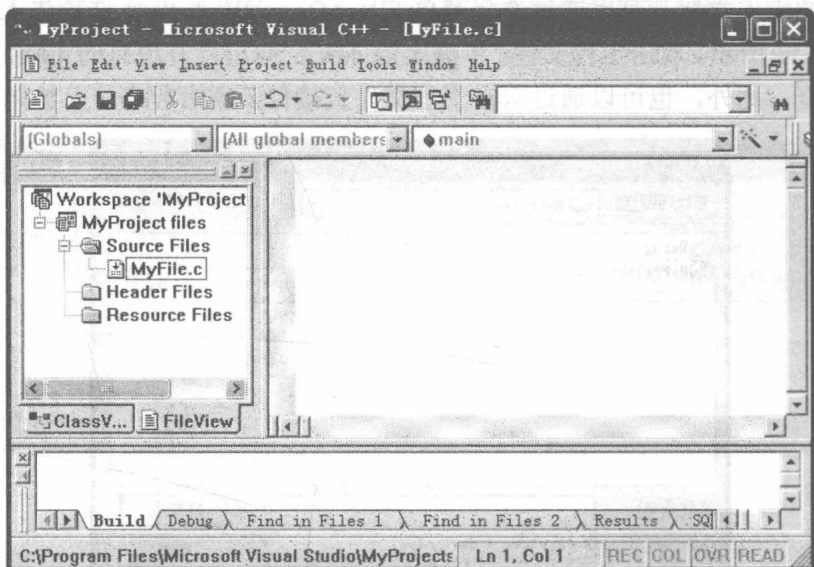


图 0.9 MyFile.c 编辑窗口

此时，程序编辑窗口被激活，可以输入和编辑源程序了。这里编写了一个简单的 C 程序，输出字符串“Hello World!”，如图 0.10 所示。

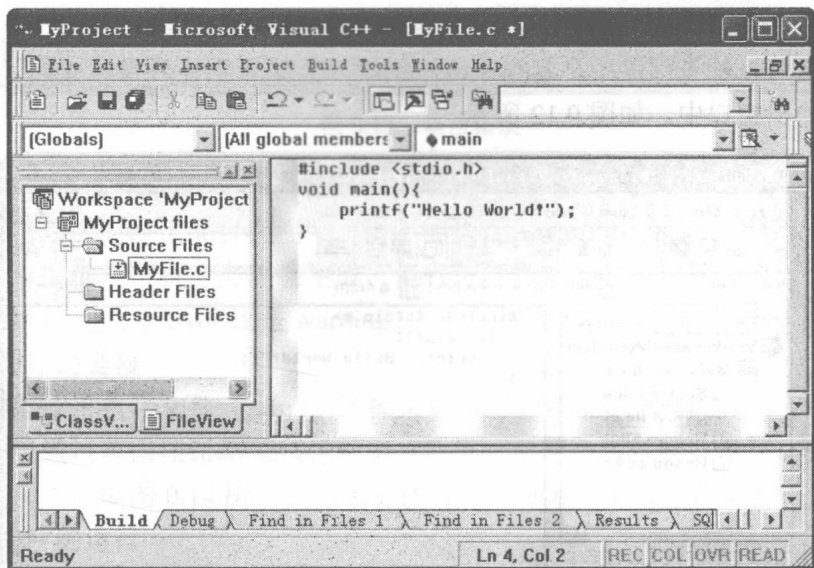


图 0.10 MyFile.c 的编辑

进入编辑状态后，如果对源程序进行了修改且未保存，则在标题栏中文件名后面会出现“*”符号提示。选择菜单 File→Save 选项，或者单击按钮，或者按快捷键 Ctrl+S，都可以保存文件。保存之后，标题栏中的“*”符号消失。

如果不想将源程序保存到指定的文件中,可以选择菜单 File→Save As 选项,重新指定文件保存的位置以及文件名。

2. 打开一个现存程序

与新建的操作类似,可以通过选择菜单 File→Open Workspace 选项进入打开工程对话框,如图 0.11 所示。双击工程文件(扩展名为.dsw),即可打开现有的工程,进入图 0.11 所示的对话框。此外,也可以通过双击工程文件直接打开现有程序。

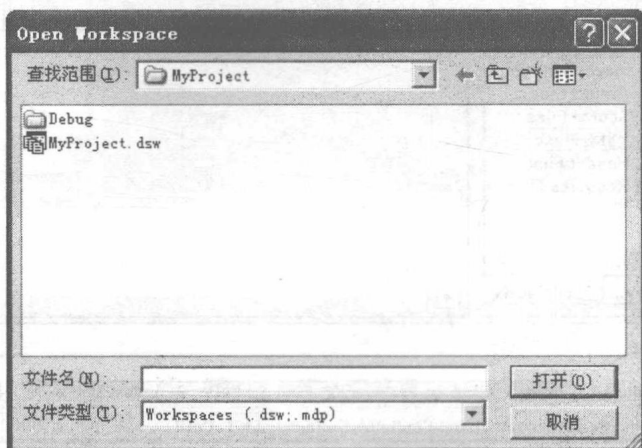


图 0.11 打开工程对话框

0.1.3 编译、链接和运行 C 程序

1. C 程序的编译

利用编译可以检查程序中是否存在语法错误,无错误则生成目标文件(.obj),编译结果显示在输出窗口中,如图 0.12 所示。

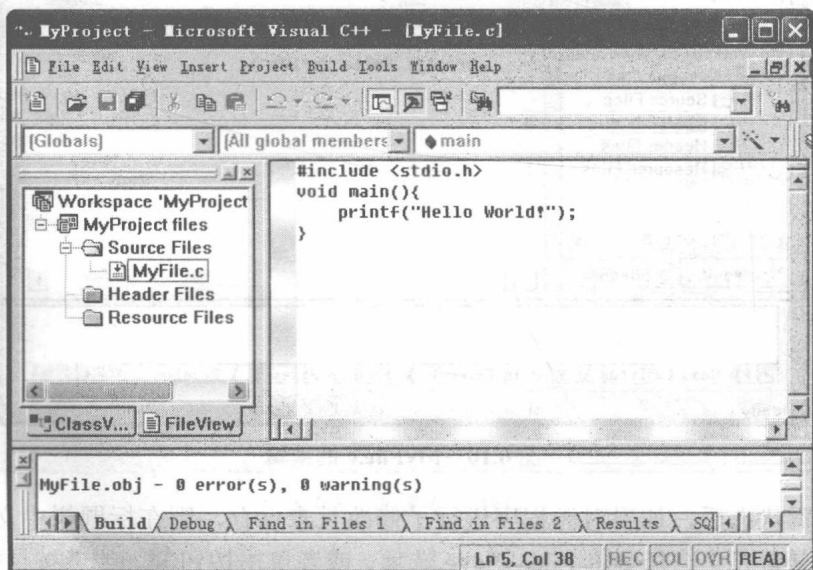


图 0.12 程序编译结果

选择菜单 Build→Compile MyFile.c 选项，即可对 MyFile.c 进行编译。此外，也可以通过单击按钮  或直接按 Ctrl+F7 键进行编译。

如果程序中存在语法错误，则可以通过双击错误提示，在程序文件中定位到错误所在的代码行。例如，如果遗漏了 printf 语句的分号，则编译出现如图 0.13 所示的结果。根据提示框中的提示信息 “syntax error: missing ‘;’ before ‘}’” 以及编辑区中的蓝色提示箭头，可以直接定位到错误处进行修改。

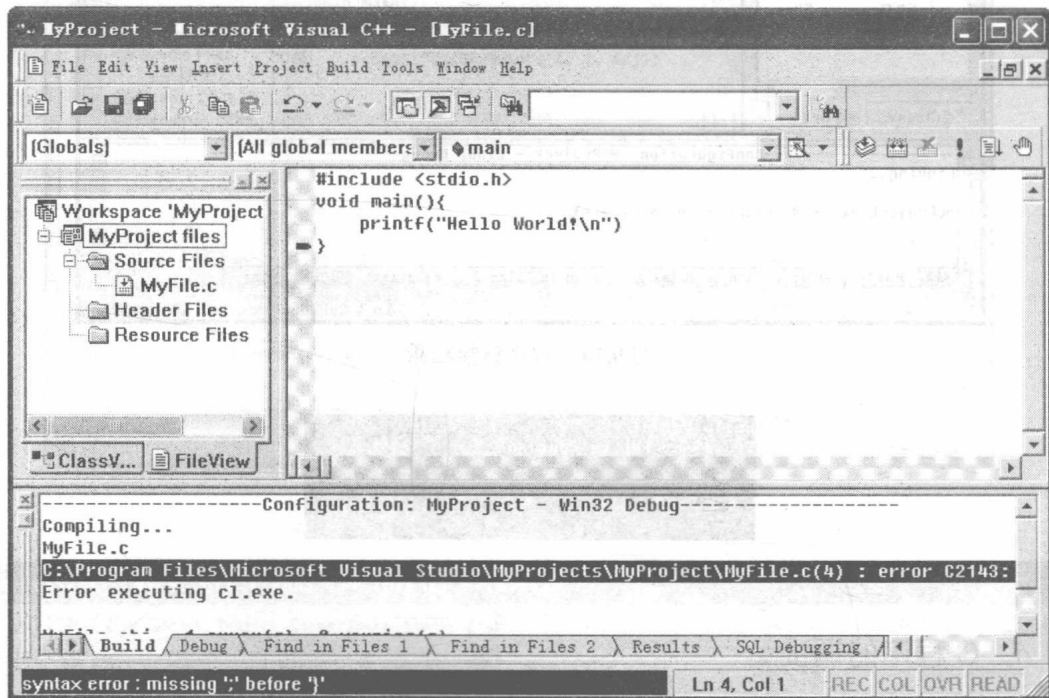


图 0.13 语法错误

值得注意的是，语法错误分为 error 和 warning 两类。error 是一类致命错误，程序中如果有此类错误则无法生成目标程序，更不能执行。warning 则是相对轻微的一类错误，不会影响目标文件及可执行文件的生成，但有可能影响程序的运行结果。因此，建议最好把所有错误（不论是 error 还是 warning）都一一修正。

2. C 程序的链接

链接将生成可执行文件（.exe）。选择菜单 Build→Build 选项可以完成程序的链接。此外，也可以通过单击按钮  或按快捷键 F7 进行链接。类似地，链接后的结果也会出现在输出窗口中，如图 0.14 所示。如果链接失败，则同样会显示出失败的具体原因。

3. C 程序的运行

选择菜单 Build→Execute MyFile.exe 选项可以启动程序的执行。此外，也可以通过单击按钮  或按快捷键 Ctrl+F5 启动程序的运行。开始运行后，将弹出一个 DOS 窗口，显示程序的运行结果，如图 0.15 所示。

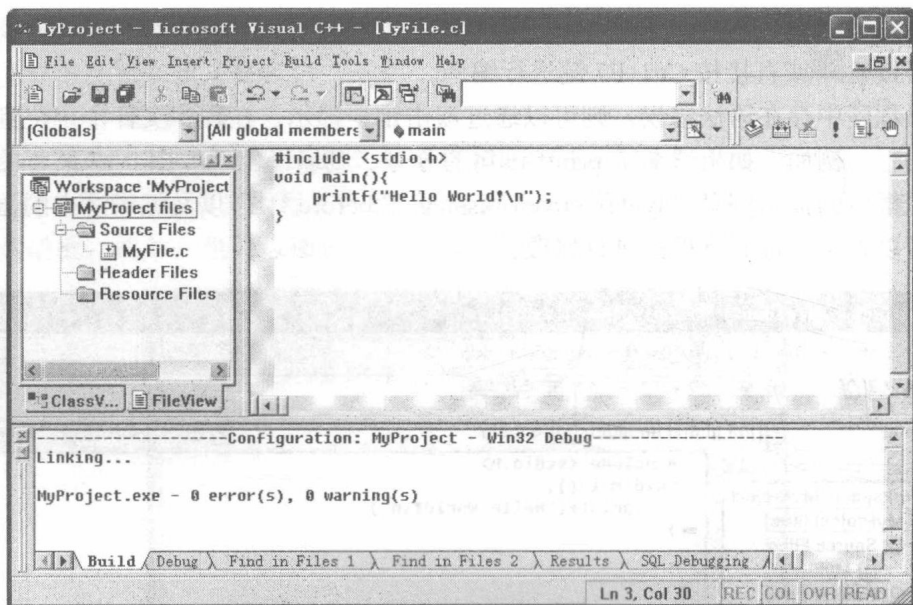


图 0.14 程序链接结果

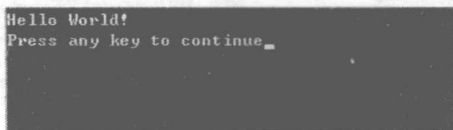


图 0.15 程序运行结果

0.1.4 C 程序的调试

调试是为了发现程序中的错误,包括语法错误和逻辑错误。其中,语法错误能够在编译的过程中发现并修改;而逻辑错误往往无法直观地被发现,即程序通常能够被成功地编译和连接,甚至执行,然而其执行结果却与预期结果不一致。逻辑错误的调试是比较困难的,因此,一般的程序开发环境都会提供完整的程序调试工具。本节将主要介绍如何使用 Visual C++ 6.0 中的调试工具发现程序中的逻辑错误。

1. 设置断点

当需要调试程序的时候,可以首先大致判断出程序中可能从哪条语句开始出现问题,并将光标移动到该语句行,单击按钮,则在语句行左侧出现一个红点,这就是断点,如图 0.16 所示。此后,程序调试运行过程中遇到断点时,将自动暂停执行,进入调试状态。

2. 进入调试界面

设置好断点之后,单击按钮,以调试的方式启动程序的运行。当运行到断点所在的语句时,将直接进入调试状态,如图 0.17 所示。代码行左侧的黄色小箭头表示了程序的当前执行位置。Debug 工具栏中有常用的调试工具按钮,表 0.1 中列举了这些常用按钮的功能。屏幕下方左右两个窗口则分别是自动变量框和观察框,可以看到变量的当前

值，作为判断程序是否出错的参考。其中，观察框支持对各种变量值的观察，在 Name 列输入变量的名字，就会在 Value 列显示出该变量的当前值。

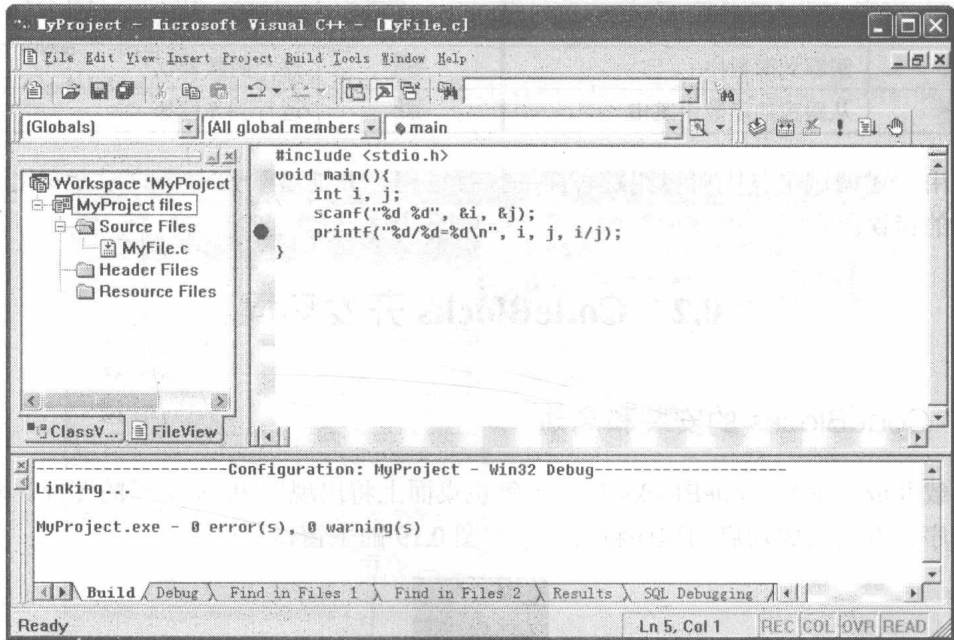


图 0.16 设置断点

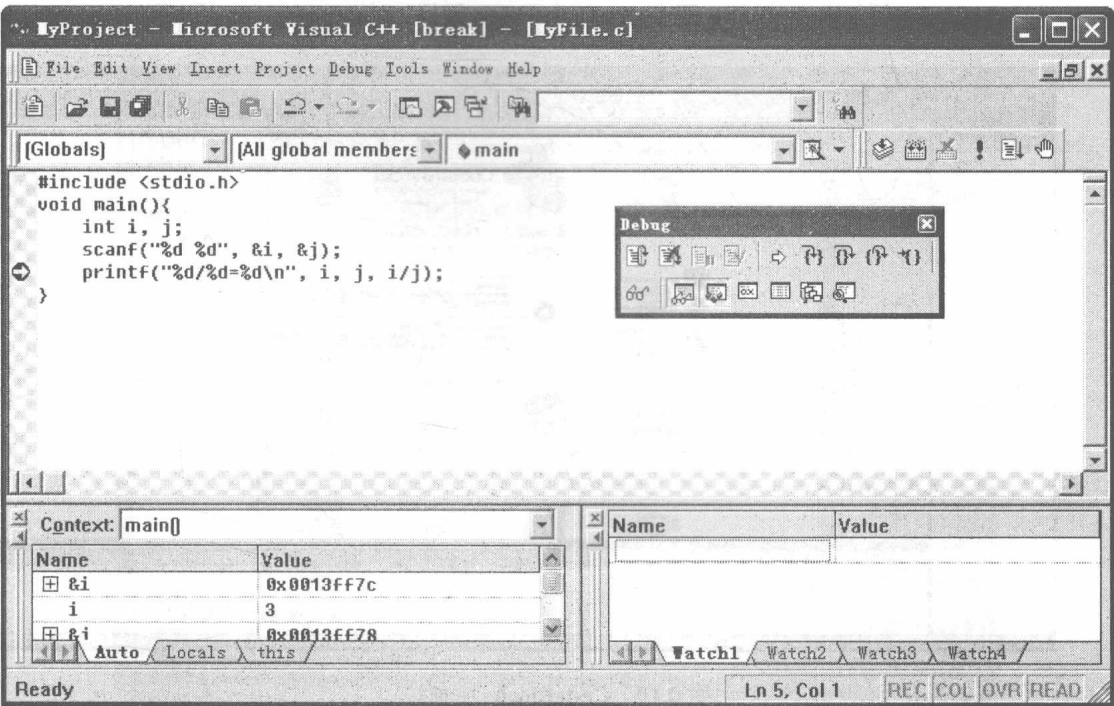


图 0.17 调试界面

表 0.1 Debug 工具栏常用按钮说明

按 钮	功 能	按 钮	功 能
	重新开始调试		停止/退出调试
	跟踪到函数中		逐个函数调试
	从跟踪的函数中跳出		运行到光标处

使用上述调试工具则可以跟踪程序的执行过程,并通过观察变量的内容变化即时发现程序的错误所在。

0.2 CodeBlocks 开发环境

0.2.1 CodeBlocks 的安装和启动

下载并成功安装 CodeBlocks 后,计算机桌面上将出现图 0.18 所示的图标。双击可运行程序。双击后成功启动程序将会显示如图 0.19 的主窗口。

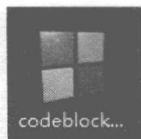


图 0.18 CodeBlocks 图标

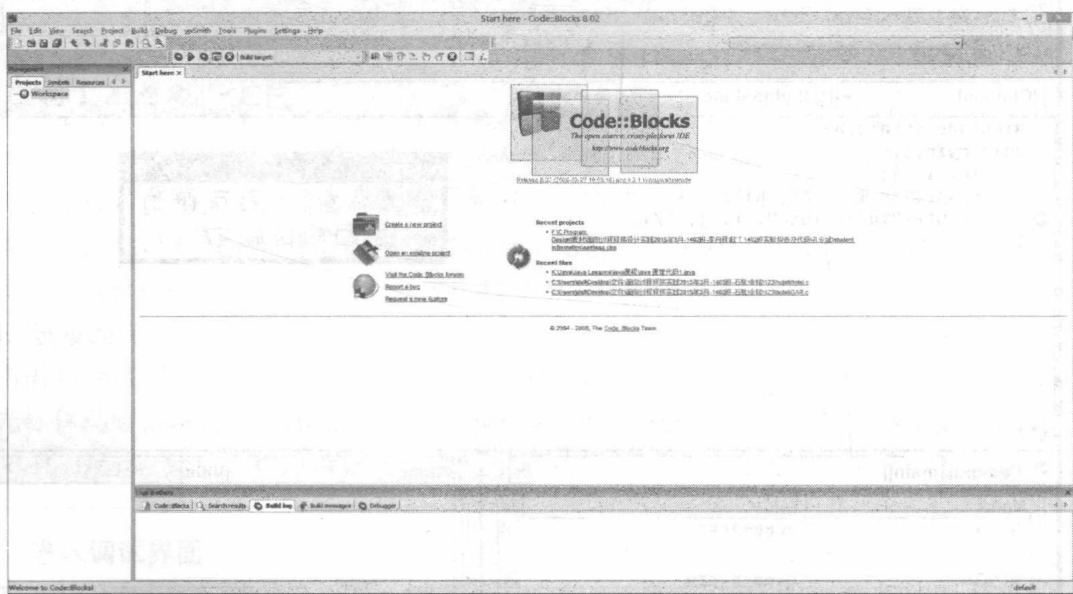


图 0.19 CodeBlocks 主窗口