



普通高等教育“十二五”规划教材 计算机应用教育系列

C语言程序设计实训教程

刘 涛 夏启寿 主编



科学出版社

普通高等教育“十二五”规划教材 计算机应用教育系列

C 语言程序设计实训教程

刘 涛 夏启寿 主 编

丁亚涛 程一飞 副主编

张 义 王 勇

科学出版社

北 京

内 容 简 介

全书共分为4章,第1章为C语言程序的调试与运行,重点介绍了 Visual C++ 6.0 和 Turbo C 2.0 集成环境下调试与运行 C 语言程序的方法。第2章为例题详解及实验指导,按章节给出了要点导读、经典例题分析、习题与实验指导。第3章是综合实训,详细给出了综合实训的目的、要求和综合实训的过程与内容。第4章是考试指导,介绍了全国高等学校(安徽考区)二级 C 语言考试和全国计算机等级考试二级 C 语言考试的背景及考试内容,给出了笔试和机试模拟卷、样卷及参考答案等。附录 I ~ III 给出了常用字符与 ASCII 代码对照表、C 语言运算符的优先级与结合性、C 语言常见错误信息表等。

本书既可作为高等院校本、专科学子学习 C 语言程序设计课程的辅助教材,也可作为 C 语言自学者的教材或参考书,同时可作为全国计算机等级考试或水平考试 C 语言考试的教材或参考书。

图书在版编目(CIP)数据

C 语言程序设计实训教程/刘涛,夏启寿主编. —北京:科学出版社,2012
(普通高等教育“十二五”规划教材 计算机应用教育系列)

ISBN 978-7-03-034320-8

I. ①C… II. ①刘… ②夏… 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(2012)第 096560 号

责任编辑:赵丽欣 郭丽娜/责任校对:耿 耘

责任印制:吕春珉/封面设计:北京科地亚蓝图文设计有限公司

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>

北京路局票据印刷厂印刷

科学出版社发行 各地新华书店经销

*

2012年8月第 一 版 开本:787×1092 1/16

2013年3月第三次印刷 印张:16 1/4

字数:388 000

定价:28.00 元

(如有印装质量问题,我社负责调换<路局票据>)

销售部电话 010-62142126 编辑部电话 010-62134021

版权所有,侵权必究

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

普通高等教育“十二五”重点规划教材
“计算机应用教育系列”学术编审委员会

主 任：胡学钢

委 员：（排名不分先后，依姓氏拼音为序）

曹双喜	陈 蕴	戴小平	胡宏智	李 鸿
李敬兆	梁五七	刘 涛	刘九才	毛 平
潘正高	钱 鸣	钱 瑛	钦明皖	秦 峰
宋万千	王 浩	王 巍	王 云	王戴木
王诗兵	吴 敏	吴 璞	夏启寿	徐 辉
许 勇	严 楠	袁鸿燕	张先宜	赵正平
郑 诚	周鸣争			

前 言

本书是由科学出版社组织多所高校教师和专家组成的编委会，根据教育部高等学校计算机科学与技术专业教学指导委员会编制的《关于进一步加强高等学校计算机基础教学的意见暨计算机基础课程教学基本要求》组织编写的系列教材之一。

本书是为配合 C 语言程序设计课程教学而编写，同时也为适应 C 语言程序设计考试的需要而精心设计的，目的是通过单元训练与综合实训，培养学生实际分析问题、动手编程能力，帮助学生系统地掌握该门课程，并对学生参加计算机等级考试和水平考试进行了指导。第 1 章为 C 语言程序的调试与运行，重点介绍了 Visual C++ 6.0 和 Turbo C 2.0 集成环境下调试与运行 C 语言程序的方法。第 2 章为例题详解及实验指导，按章节给出了要点导读与经典例题分析供学习者参考。并精心挑选了适量的习题，包括选择题、填空题和编程题，供读者练习；同时，按照教学进度每章安排一至两个实验，供教学选用。第 3 章是综合实训，详细给出了综合实训的目的、要求和过程。实训目的与要求明确、过程清晰、内容丰富，可作为 C 语言课程设计的参考内容。第 4 章是考试指导，参考了最新的全国高等学校计算机水平考试安徽考区 C 语言考试和全国计算机等级考试二级 C 语言考试的大纲，并给出了笔试和机试模拟卷或样卷及参考答案等。为了方便学习者参考与查阅，本书附录 I ~ III 给出了常用字符与 ASCII 代码对照表、C 语言运算符的优先级与结合性、C 语言常见错误信息表等。编者对本书所给出的习题与实验指导均作了详细的解答，由于版面有限请登录 www.abook.cn 下载。

本书由刘涛、夏启寿担任主编，丁亚涛、程一飞、张义、王勇任副主编，参与编写工作的还有范莉莉、严轶群、强俊、童莉、黄伟、赵森严等。周鸣争教授对全书进行了认真审核。胡学钢教授、王浩教授对本书提出了很多宝贵的建议。同时，在全书的策划和出版过程中，得到了许多从事 C 语言教学工作的同仁的关心和帮助，在此一并感谢。

由于编者水平有限，书中不足之处，敬请广大读者批评指正，以便我们再版时修正。

目 录

第 1 章 C 语言程序的调试与运行	1
1.1 C 语言程序调试与运行过程及错误分类	1
1.1.1 C 语言程序调试与运行的一般过程	1
1.1.2 C 语言程序调试与运行中常见错误分类	2
1.2 Visual C++ 6.0 环境下程序的调试与运行	2
1.2.1 Visual C++ 6.0 的安装和启动	2
1.2.2 Visual C++ 6.0 常用功能介绍	3
1.2.3 Visual C++ 6.0 工具栏的使用	4
1.2.4 建立和执行包含一个文件的 C 程序	5
1.2.5 建立和运行包含一个文件的 C++ 程序	12
1.2.6 建立和执行包含多个文件的程序	14
1.3 Turbo C 2.0 环境下程序的调试与运行	25
1.3.1 Turbo C 2.0 的安装	25
1.3.2 Turbo C 2.0 集成开发环境介绍	25
1.3.3 C 语言程序的编辑、编译和运行	29
1.4 撰写 C 语言程序调试报告的参考格式	34
第 2 章 例题详解及实验指导	36
2.1 C 语言概述	36
2.1.1 要点导读	36
2.1.2 经典例题分析	38
2.1.3 习题	39
2.2 数据类型与表达式	39
2.2.1 要点导读	39
2.2.2 经典例题分析	41
2.2.3 习题	44
2.2.4 实验指导	45
2.3 顺序结构程序设计	47
2.3.1 要点导读	47
2.3.2 经典例题分析	47
2.3.3 习题	51
2.3.4 实验指导	53
2.4 选择结构程序设计	55
2.4.1 要点导读	55

2.4.2	经典例题分析	56
2.4.3	习题	58
2.4.4	实验指导	61
2.5	循环结构程序设计	62
2.5.1	要点导读	62
2.5.2	经典例题分析	63
2.5.3	习题	65
2.5.4	实验指导	69
2.6	数组	71
2.6.1	要点导读	71
2.6.2	经典例题分析	72
2.6.3	习题	76
2.6.4	实验指导	79
2.7	函数与预处理	81
2.7.1	要点导读	81
2.7.2	经典例题分析	86
2.7.3	习题	92
2.7.4	实验指导	98
2.8	指针	102
2.8.1	要点导读	102
2.8.2	经典例题分析	106
2.8.3	习题	110
2.8.4	实验指导	114
2.9	结构体与共用体	117
2.9.1	要点导读	117
2.9.2	经典例题分析	119
2.9.3	习题	122
2.9.4	实验指导	125
2.10	位运算和文件	128
2.10.1	要点导读	128
2.10.2	经典例题分析	129
2.10.3	习题	137
2.10.4	实验指导	139
2.11	C++程序设计基础	140
2.11.1	要点导读	140
2.11.2	经典例题分析	141
2.11.3	习题	144
2.11.4	实验指导	145

第3章 综合实训	147
3.1 综合实训的目的	147
3.2 综合实训的要求	148
3.3 综合实训的过程	148
3.4 综合实训的内容	149
第4章 考试指导	167
4.1 全国高等学校（安徽考区）二级 C 语言考试指导	167
4.1.1 考试介绍及分析	167
4.1.2 二级 C 语言笔试模拟卷及参考答案	170
4.1.3 上机考试模拟卷及参考答案	201
4.2 全国计算机等级考试二级 C 语言考试指导	222
4.2.1 考试介绍及分析	222
4.2.2 NCRE 二级 C 语言笔试样卷及参考答案	226
4.2.3 NCRE 二级 C 语言上机考试样卷及参考答案	235
附录 I 常用字符与 ASCII 码对照表	242
附录 II C 语言运算符的优先级与结合性	244
附录 III C 语言常见错误信息表	245
参考文献	249

第 1 章 C 语言程序的 调试与运行

1.1 C 语言程序调试与运行过程及错误分类

1.1.1 C 语言程序调试与运行的一般过程

在计算机上编辑和运行一个 C 程序要经过以下几个步骤：输入与编辑源程序→对源程序进行编译→与库函数连接→运行可执行的目标程序，如图 1-1 所示。

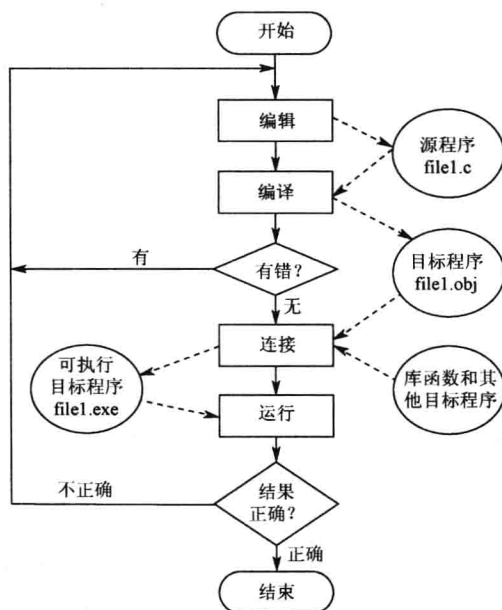


图 1-1 C 语言程序的调试与运行过程

图 1-1 中实线表示操作流程，虚线表示文件的输入/输出。例如，编辑后得到一个源程序文件 file1.c，然后对源程序文件 file1.c 进行编译，经过编译得到目标程序文件 file1.obj，再将目标程序文件 file1.obj 与系统提供的库函数等连接，得到一个可执行的目标程序文件 file1.exe，最后运行 file1.exe，直至得到正确结果。

1.1.2 C 语言程序调试与运行中常见错误分类

C 源程序可以在 Turbo C 或 Visual C++ 等集成环境中进行编译、连接和运行。在调试与运行过程中,可能会出现错误。C 语言常见错误分为 4 类。

(1) 语法或编译出错,编译后给出两种提示。警告(warning):非致命错误,一般不影响运行;错误(error):需改正后重新编译、连接才能运行。通常情况下我们应该修改程序直至既无“错误(error)”也无“警告(warning)”性错误。

(2) 连接出错,编译正确后,连接也可能出现错误。一般是函数名称错,需改正后重新编译连接;或调用系统函数出错,需检查选项菜单的目录设置。例如,在 Turbo C 环境下,某台机器上其集成环境安装在 D:盘,但是选项菜单的目录设置上却是 C:盘,这时需通过菜单进行重新设置;Visual C++ 环境下,两个程序在同个工作空间运行,有时会导致连接错误,这时退出集成环境重新进入即可。

(3) 运行错误:编译、连接正常,但不能正常运行。原因可能是程序中有死循环、除数为 0、地址运用错误、数据类型与格式控制不匹配等错误。

(4) 逻辑错:能运行,但结果不是期望的结果。原因主要是输入数据错或程序逻辑等错误。

下面介绍在 Visual C++ 6.0 和 Turbo C 2.0 集成环境中程序的调试与运行过程。

1.2 Visual C++ 6.0 环境下程序的调试与运行

Visual C++ 6.0(简称 VC++ 6.0)是美国 Microsoft 公司开发的 Visual Studio 的一部分,是一个使用广泛的 C++ 集成开发环境。VC++ 6.0 集程序代码的编辑、编译、连接、调试于一体,给编程人员提供了一个完整、方便的开发界面和许多有效的辅助开发工具。VC++ 6.0 的编辑环境包含了许多独立的组件,所有这些组件的功能都隐藏在菜单和工具条中。

1.2.1 Visual C++ 6.0 的安装和启动

现在常用的是 Visual C++ 6.0 版本,有英文版和汉化版。其实,Visual C++ 的不同版本的上机操作方法大同小异,掌握了其中的一种,就能举一反三地使用其他版本。这里以 Visual C++ 6.0 英文版为背景来介绍 C 语言程序在 Visual C++ 环境下的上机调试与执行过程。

如果使用的计算机没有安装 Visual C++ 6.0,则应先安装 Visual C++ 6.0。Visual C++ 是 Visual Studio 的一部分,因此需要找到 Visual Studio 的光盘,执行其中的 setup.exe,并按屏幕上的提示进行安装。安装结束后,在 Windows 的“开始”→“程序”子菜单中就会出现 Microsoft Visual Studio 子菜单。

需要使用 Visual C++ 时,只需从桌面上顺序选择“开始”→“程序”→Microsoft Visual Studio→Visual C++ 6.0 即可。此时屏幕在短暂显示 Visual C++ 6.0 的版权页后,出现 Visual C++ 6.0 的主窗口,如图 1-2 所示。

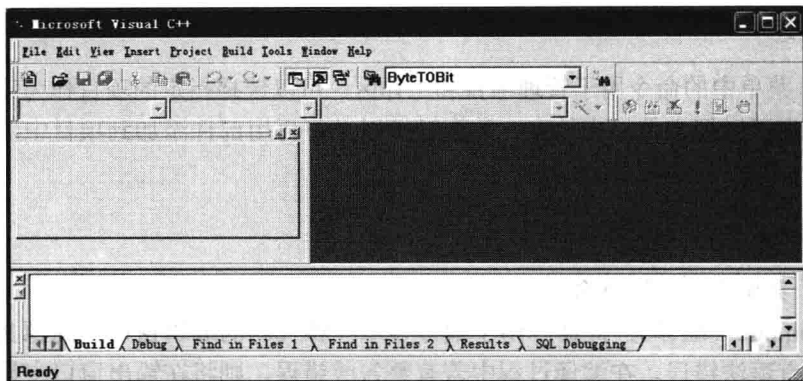


图 1-2 Visual C++ 6.0 的主窗口

也可以先在桌面上建立 Visual C++ 6.0 的快捷方式图标, 需要使用 Visual C++ 时只需要双击桌面上的该图标即可, 此时屏幕上同样会弹出如图 1-2 所示的 Visual C++ 主窗口。

1.2.2 Visual C++ 6.0 常用功能介绍

在 Visual C++ 主窗口的上部有 Visual C++ 的主菜单栏, 其中包含 9 个菜单项: File (文件)、Edit (编辑)、View (查看)、Insert (插入)、Project (项目)、Build (构建)、Tools (工具)、Window (窗口) 和 Help (帮助)。

主窗口的左侧是项目工作区窗口, 右侧是程序编辑窗口。工作区窗口用来显示所设定的工作区的信息, 程序编辑窗口用来输入和编辑源程序。

1. “File (文件)” 菜单

“文件”菜单中共有 14 个菜单项, 分成 6 组, 组与组之间通过凹线分隔开。第一组是基本文件操作; 第二组是工作区 (Workspace) 操作, 工作区是 VC++ 中最大的一个处理单位, 每个工作区包括多个工程, 每个工程又包含若干个文件; 第三组用于文件保存; 第四组用于文件打印; 第五组显示新近打开的文件和工作区; 第六组退出 VC++ 环境。

2. “Edit (编辑)” 菜单

“编辑”菜单包含用于编辑和搜索的命令, 这些命令与其他 Windows 应用程序中的同名命令具有相似的功能。

3. “View (视图)” 菜单

“视图”菜单包含用于检查源代码和调试信息的命令项, 可以用来设置全屏方式显示窗口, 或者打开工作区窗口、输出窗口和各种调试窗口。

4. “Insert (插入)” 菜单

“插入”菜单可以创建新的类、资源窗体并将它们插入到文档中, 可以将文件作为文本插入到文件中, 也可以添加新的对象到项目中。

5. “Project (工程)” 菜单

“工程”菜单中的命令用于管理项目和工作区。可以选择指定的项目作为工作区中的活动项目,也可以把文件、文件夹、数据链接以及可再用部件添加到项目中,还可以编辑和修改项目间的依赖关系。

6. “Build (构建)” 菜单

“构建”菜单中包括以下命令。

(1) **Compile (编译)**: 此命令用于编译显示在源代码的编辑窗口中的源文件并检查源文件是否有语法错误。在编译过程中若有警告或错误,则将在输出窗口中显示错误信息。可以向前或向后浏览输出窗口中的错误信息,然后按 F4 键在源代码编辑窗口中显示相应的代码行。

(2) **Build (构建)**: 即连接,用于创建项目的可执行目标文件,但不运行它。

(3) **Build all (重建全部)**: 重新编译所有文件(包括资源文件),重新连接生成可执行的目标文件。

(4) **Batch Build (批构建)**: 成批编译、连接不同项目或同一项目的不同设置。

(5) **Clean (清除)**: 把编译、连接过程中所生成的中间文件和最终可执行的目标文件删除掉。

(6) **Start Debug (开始调试)**: 在调试状态下运行程序,程序运行到一个断点处暂时停止。

(7) **Execute (执行)**: 运行可执行目标文件,如果此文件比源代码旧,则首先编译项目,再运行新产生的可执行目标文件。

7. “Tools(工具)” 菜单

“工具”菜单主要实现启动常用集成开发环境常用工具栏的功能,提供一些 Visual C++6.0 的实用工具,包括浏览程序符号、定制菜单与工具栏、激活常用工具等。

8. “Windows(窗口)” 菜单

“窗口”菜单分成 4 组菜单命令,用于操作 IDE 的窗口,包括窗口设置命令、窗口翻页命令、窗口摆放命令和窗口资源命令。

9. “Help (帮助)” 菜单

“帮助”菜单主要用于集成开发环境的帮助操作,如启动 MSDN、版本说明等。

1.2.3 Visual C++ 6.0 工具栏的使用

Visual C++ 6.0 默认显示常用工具栏和编译微型条工具栏,如图 1-3 所示。

常用工具栏按钮大部分与 Word 相同,前面几个按钮的功能如下:

- ① **New Text File**: 建立新文本文件;
- ② **Open (Ctrl+O)**: 打开文件;

③ Save (Ctrl+S): 保存文件;

④ Save All: 保存所有文件。

编译微型条工具栏按钮的功能如下:

⑤ Compile (Ctrl+F7): 编译源程序文件形成.OBJ 目标代码文件;

⑥ Build (F7): 构建(编译连接)形成.EXE 可执行文件;

⑦ BuildStop (Ctrl+Break): 停止编译连接;

⑧ BuildExecute (Ctrl+F5): 运行可执行文件(若未形成可执行文件可先连接), 运行后按任意键返回集成环境;

⑨ Go (F5): 调试执行(可以建立工作区、编译、连接直至运行), 运行后直接返回集成环境;

⑩ Insert/Remove Breakpoint (F9): 插入/删除断点。

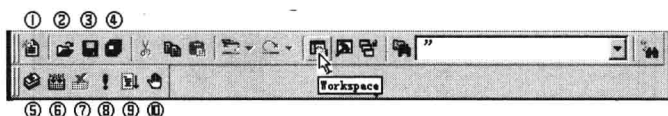


图 1-3 常用工具栏(上)和编译微型条工具栏(下)

1.2.4 建立和执行包含一个文件的 C 程序

C 源程序可以在 Visual C++ 集成环境中进行编译、连接和运行。例 1.1 给出了最简单的情况, 程序只由一个源程序文件组成, 即单文件程序。

【例 1.1】从键盘输入两个整数, 求这两个数之差。

```
/*FILE1_1.C*/  
#include <stdio.h>  
void main()  
{  
    int a,b,c;  
    printf("Please input a,b: ");  
    scanf("%d,%d",&a,&b);  
    c=a-b;  
    printf("c=%d\n",c);  
}
```

1. 建立、编辑源程序

1) 新建并编辑一个源程序

在 Visual C++ 主窗口的主菜单栏中单击“File (文件)”, 然后在其下拉菜单中单击“New (新建)”。

屏幕上出现一个“New (新建)”对话框, 如图 1-4 所示。单击此对话框上方的“Files (文件)”标签, 在其下拉菜单中有“C++ Source File”项, 表示这项的功能是建立新的 C++ 源程序文件。Visual C++ 6.0 既可以用于处理 C++ 源程序, 也可以用于处理 C 源程序, 因此选择 C++ Source File。然后在对话框右半部分的“Location (目录)”文本框中输入

准备编辑的源程序文件的存储路径（假设为 D:\VC），表示准备编辑的源程序文件将存放在 D:\VC 子目录下。在其上方的“File（文件）”文本框中输入准备编辑的源程序文件的名字（假设为 FILE1_1.C），表示要建立的是 C 源程序，即将进行输入和编辑的源程序就以 FILE1_1.C 为文件名存放在 D 盘的 VC 目录下。当然，也可以指定其他的路径名和文件名。

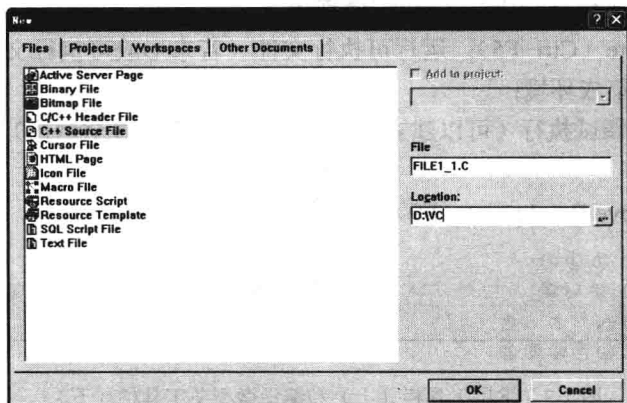


图 1-4 新建 C 源程序对话框

注意

这里指定的文件名后缀为.C，如果输入的文件名为 FILE1_1.CPP，则表示要建立的是 C++ 源程序。如果不写后缀，系统会默认指定为 C++ 源程序文件，自动加上后缀.CPP。

在单击“OK(确定)”按钮后，回到 Visual C++ 主窗口，由于在前面已指定路径(D:\VC)和文件名(FILE1_1.C)，因此在窗口的标题栏中显示出 D:\VC\FILE1_1.C。可以看到光标在程序编辑窗口闪烁，表示程序编辑窗口已激活，这样就可以输入和编辑源程序了。输入例 1.1 所示的程序，如图 1-5 所示。在输入过程中可能会出现一些编辑错误，如用户能及时发现问题，可以进行修改。

在图 1-5 的最下部的中间显示了“Ln 10, Col 2”，表示光标当前的位置在第 10 行第 2 列，当光标位置改变时显示的数字也随之改变。

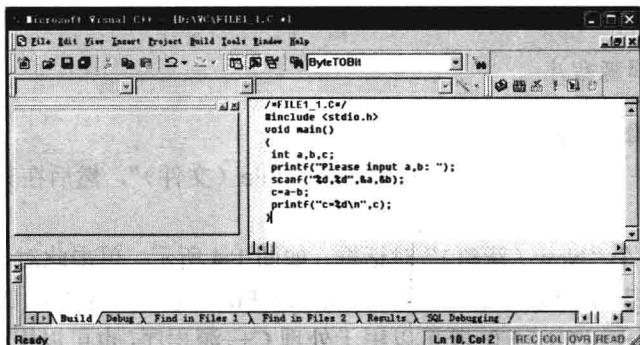


图 1-5 编辑窗口

如果经检查无误,则将源程序保存在前面指定的文件中,方法是:在主菜单栏中选择“File(文件)”,并在其下拉菜单中选择“Save(保存)”项,如图1-6所示。也可以用快捷键 Ctrl+S 或单击工具栏上的 Save 按钮来保存。如果保存后又发现了错误,应该先修改程序,然后再用上述方法保存。

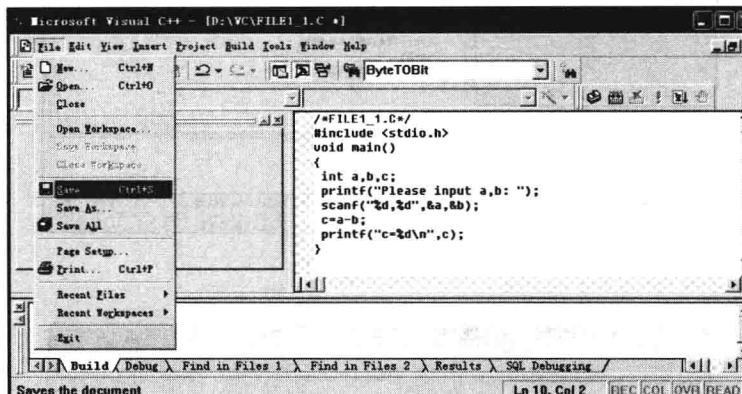


图 1-6 保存窗口

如果不想将源程序存放到原先指定的文件中,则选择“Save As(另存为)”项,并在弹出的“Save As(另存为)”对话框中指定文件路径和文件名。

2) 通过已有的程序建立并编辑一个新程序

如果已经编辑并保存过 C 源程序,则可以通过一个已有的程序来建立一个新程序,这样利用原有程序中的部分内容比重重新输入一个新文件方便,方法如下:

① 打开任何一个已有的源文件(如 FILE1_1.C)。

② 利用该文件修改成新的文件,然后通过菜单“File(文件)”→“Save As(另存为)”将它以另一文件名保存(如以文件名“FILE1_2.C”另存),这样就生成了一个新文件 FILE1_2.C。

用这种方法很方便,但应注意在保存新文件时,不要用“File(文件)”→“Save(保存)”操作,否则原有文件(FILE1_1.C)的内容就被修改了。

2. 编译、调试和连接

1) 程序的编译

在编辑和保存了源文件(如 FILE1_1.C)后,若需要对该源文件进行编译,单击主菜单栏中的“Build(构建)”,在其下拉菜单中选择“Complie FILE1_1.C(编译 FILE1_1.C)”项,如图1-7所示。也可以不通过单击菜单,而用快捷键 Ctrl+F7 或者单击工具栏中的编译图标来实现程序的编译。由于建立(或保存)文件时已指定了源文件的名称 FILE1_1.C,因此在 Build 菜单的 Compile 项中就自动显示了当前要编译的源文件名 FILE1_1.C。

在单击编译命令后,屏幕上出现一个对话框,内容是“This build command requires an active project workspace. Would you like to create a default project workspace?”(此编译命令要求一个有效的项目工作区,你是否同意建立一个默认的项目工作区),如图1-8所示。单击“是(Y)”按钮,表示同意由系统建立默认的项目工作区。

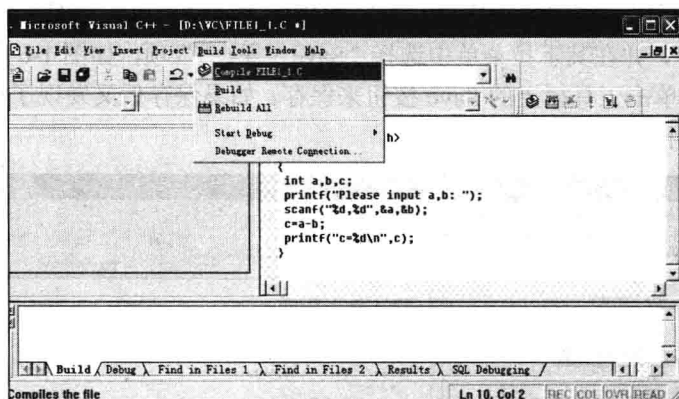


图 1-7 编译窗口

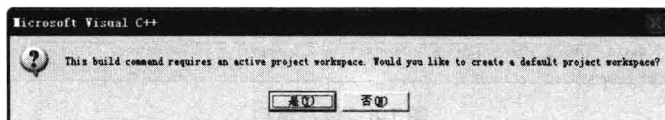


图 1-8 编译前对话框 (a)

单击图 1-8 中的“是 (Y)”则开始编译；但是如果编译前没有保存，则弹出对话框，如图 1-9 所示。单击“是 (Y)”，然后开始编译。



图 1-9 编译前对话框 (b)

在进行编译时，编译系统检查源程序中有无语法错误，然后在主窗口下部的调试信息窗口输出编译的信息，如图 1-10 所示。如果有错，就会指出错误的位置和性质。

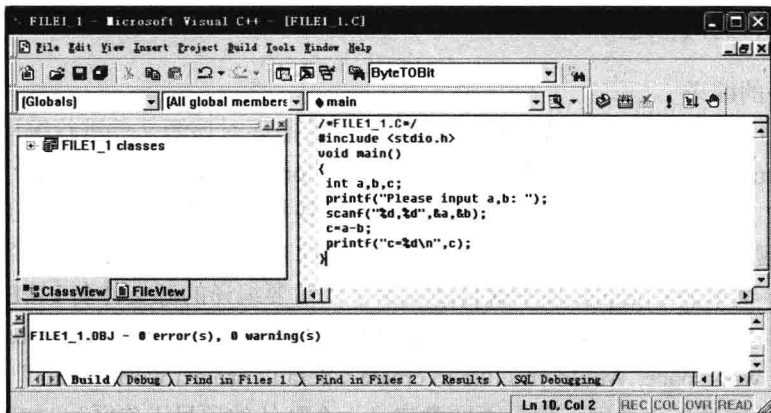


图 1-10 编译后窗口 (a)

2) 程序的调试

一般来说,我们编写的程序可能存在错误。调试的任务是发现和改正程序中的错误,使程序能正常运行。编译系统能检查出程序中的语法错误,语法错误分为两类:一类是致命错误,以 **error** 表示,如果程序中有这类错误就通不过编译,无法形成目标程序,更谈不上运行了;另一类是轻微错误,以 **warning** 表示,这类错误不影响生成目标程序,但有可能影响运行的结果。因此也应当改正,使程序既无 **error**,又无 **warning**。

若在输入例 1.1 的程序时,第 2 个 **printf** 语句丢了分号,编译后,如图 1-11 所示。

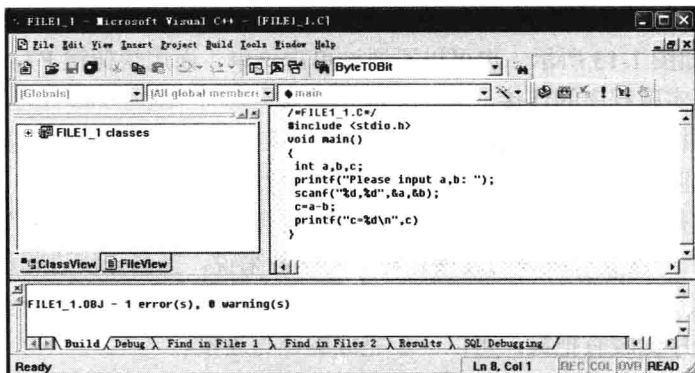


图 1-11 编译后窗口 (b)

在图 1-11 中的调试信息窗口中可以看到编译的信息,指出源程序有 1 个 **error**。用鼠标单击调试信息窗口中右侧的向上箭头,可以看到出错的位置和性质,如图 1-12 所示。

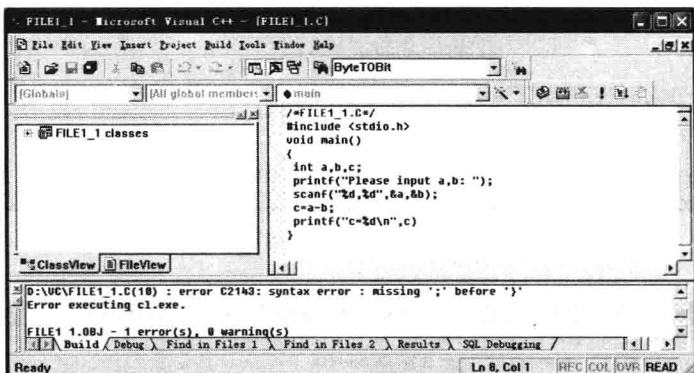


图 1-12 编译后窗口 (c)

从图 1-12 下部调试信息窗口所示的信息中可以看到,第 10 行有致命错误,错误的性质是: **syntax error : missing ';' before '}'** (语法错误:在'}'前丢失了';')。

检查图 1-12 中的程序,发现第 9 行末少了一个分号。这里要注意:大家觉得应该是第 9 行有错,怎么在报错时说成是第 10 行有错呢?这是因为 C 语言允许将一个语句分写成几行,因此检查完第 9 行末尾无分号时还不能判定该语句有错,必须再检查下一行,直到发现第 10 行的“}”前没有分号(;),才判定出错。因此报错在第 10 行。在分析编译报错信息时,应该检查出错点的上下行,一个错误也可能影响程序的其他地方,所以