

普通高等教育
“十二五”规划教材



Experimental Guide of C Programming Language

C语言程序设计 实验教程

刘成忠 王联国 主编

中国林业出版社

普通高等教育“十二五”规划教材

C 语言程序设计实验教程

刘成忠 王联国 主编

中国林业出版社

内 容 简 介

本教材可帮助学生进一步消化和吸收 C 语言的相关知识,更好地掌握 C 语言的编程技能,提高运用 C 语言解决实际问题的能力。

全书包括 3 章正文内容和 1 项附录(《C 语言程序设计教程》(第二版)习题解答):第 1 章为 Visual C++6.0 集成开发环境简介,主要介绍了 Visual C++6.0 工作环境的建立及使用;第 2 章为上机实验项目,包括了 17 项验证性实验和 1 项综合性(设计性)实验;第 3 章为模拟试题及参考答案,主要围绕全国计算机等级考试(二级 C 语言)无纸化考试的要求,使学生更快更好地熟悉和适应无纸化考试题型。

图书在版编目(CIP)数据

C 语言程序设计实验教程 / 刘成忠,王联合国主编. —北京:中国林业出版社,2015.3

普通高等教育“十二五”规划教材

ISBN 978-7-5038-7821-3

I. ①C… II. ①刘… 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 005019 号

中国林业出版社·教育出版分社

责任编辑:张东晓

电话:(010) 83143560 83143551

传真:(010) 83143516

出版发行 中国林业出版社(100009 北京市西城区德内大街刘海胡同 7 号)

E-mail: jiaocaipublic@163.com 电话:(010) 83143500

http://lycb.forestry.gov.cn

经 销 新华书店

印 刷 北京市昌平百善印刷厂

版 次 2015 年 3 月第 1 版

印 次 2015 年 3 月第 1 次印刷

开 本 850mm×1168mm 1/16

印 张 20.25

字 数 517 千字

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

《C 语言程序设计实验教程》

编写人员

主 编：刘成忠 王联合国

编写人员：（按姓氏笔画排序）

王联合国 刘立群 刘成忠

刘泽华 李 玥 陈映江

韩俊英 雒翠萍

前言

本教材是《C 语言程序设计教程》(第二版)(中国农业大学出版社)的配套实验教材。全书共有 4 个部分,突出适用性、创新性、实用性三大特点,应用系统的实例去引导学生按步骤进行操作和编程,全面地学习掌握 C 语言内容体系中的各知识点和编程要点,在学习程序设计的过程中,使学生养成良好的编程风格和习惯,更快地掌握程序设计、程序调试的方法和技巧。

为配合全国计算机等级考试(二级 C 语言)无纸化考试的要求,实验教材的运行环境建立在 Visual C++ 6.0 集成开发环境下。本书对 Visual C++ 6.0 开发环境进行了详细的介绍,使学生能尽快掌握编程环境。为了让学生利用所学知识解决相关的实际问题,从实际需求出发,设计程序实例;同时,为了让学生更快更好地熟悉和适应无纸化考试的题型,实验教材中编写了 16 套无纸化模拟试题,包括选择题、程序改错题、程序填空题和程序设计题 4 种题型。

本教材题型多样、题量丰富、由简到难、深入浅出,既注重了理论知识的强化,又强调了实践技能的培养,既可作为各类高等院校和相关培训机构 C 语言程序设计课程的教学参考书,也可作为各类考生参加全国计算机等级考试(二级 C 语言)的应试指导用书。

在编写过程中,许多老师对本书内容的组织和安排提出了很多有益的建议,中国林业出版社为本教材的出版提供了大力支持和帮助,我们对此表示由衷的感谢和敬意。

由于编者水平有限,时间仓促,不足与欠妥之处在所难免,衷心希望广大读者批评指正。

刘成忠 王联国

2014 年 10 月

目 录

第 1 章	Visual C++ 6.0 集成开发环境简介	(1)
1.1	Visual C++ 6.0 工作环境介绍	(1)
1.2	Visual C++ 6.0 环境中建立 C 语言源程序	(5)
1.3	源程序的编译、连接和运行	(9)
1.4	程序的动态调试	(11)
1.5	退出 Visual C++ 6.0 工作环境	(19)
第 2 章	上机实验项目	(20)
实验一	熟悉 Visual C++ 6.0 工作环境	(20)
实验二	数据类型、运算符与表达式	(21)
实验三	顺序结构程序设计	(24)
实验四	选择结构程序设计	(26)
实验五	循环结构程序设计(一)	(33)
实验六	循环结构程序设计(二)	(40)
实验七	数组(一)	(43)
实验八	数组(二)	(47)
实验九	函数(一)	(51)
实验十	函数(二)	(54)
实验十一	函数(三)	(56)
实验十二	指针(一)	(60)
实验十三	指针(二)	(66)
实验十四	指针(三)	(70)
实验十五	结构体、共用体和枚举类型	(75)
实验十六	文件	(79)
实验十七	面向对象程序设计 C++	(82)
实验十八	综合性(设计性)实验	(92)
第 3 章	模拟试题及参考答案	(100)
模拟试题(一)		(100)
模拟试题(二)		(111)
模拟试题(三)		(123)
模拟试题(四)		(134)
模拟试题(五)		(144)

模拟试题(六)	(154)
模拟试题(七)	(164)
模拟试题(八)	(175)
模拟试题(九)	(185)
模拟试题(十)	(199)
模拟试题(十一)	(210)
模拟试题(十二)	(221)
模拟试题(十三)	(232)
模拟试题(十四)	(244)
模拟试题(十五)	(257)
模拟试题(十六)	(269)
附录 配套教材《C 语言程序设计教程》(第二版)课后习题解答	(281)
参考文献	(315)

▶▶▶ 第 1 章 Visual C ++ 6.0 集成开发环境简介

本部分对 Visual C ++ 6.0 集成开发环境进行详细的介绍,通过实例讲解如何利用 Visual C ++ 6.0 集成开发环境建立 C 语言源程序以及对源程序进行编译、连接、运行的方法,程序动态调试的基本方法和技巧等,使学生能尽快掌握编程环境。

1.1 Visual C ++ 6.0 工作环境介绍

1.1.1 Visual C ++ 6.0 简介

Visual C ++ 6.0 是微软公司 1998 年推出的一款基于 Windows 平台的 C/C ++ 可视化集成开发环境,由于其良好的界面和可操作性,被广泛应用。Visual C ++ 6.0 是 Microsoft Visual Studio 6.0 家族成员之一,与其他的可视化编程环境(如 Visual Basic)一样,Visual C ++ 6.0 集程序的代码编辑、编译、连接、调试等功能于一体,给编程人员提供了一个完整、方便的开发环境,并提供了许多有效的辅助开发工具。Visual C ++ 6.0 还提供了功能强大的资源编辑器和图形编辑器,利用“所见即所得”的方式完成程序界面的设计,大大减轻了程序设计的劳动强度,提高了程序设计的效率。

1.1.2 Visual C ++ 6.0 的安装和启动

运行 Visual Studio 软件中的 setup.exe 程序,选择安装 Microsoft Visual C ++ 6.0,然后按照安装程序的向导提示逐步完成安装过程。安装完成后,单击任务栏中的“开始”按钮,选择“程序”菜单,在“程序”的级联菜单中就会出现“Microsoft Visual Studio 6.0”子菜单,在“Microsoft Visual Studio 6.0”的级联菜单中选择“Microsoft Visual C ++ 6.0”菜单项即可启动 Visual C ++ 6.0,也可以通过双击桌面快捷方式启动 Visual C ++ 6.0。

1.1.3 Visual C ++ 6.0 集成化操作界面

Visual C ++ 6.0 启动完成后,即可出现其主窗口,如图 1-1 所示。

Visual C ++ 6.0 主窗口的顶部包括标题栏、菜单栏和工具栏,其中菜单栏包括 9 个菜单项:File(文件)、Edit(编辑)、View(查看)、Insert(插入)、Project(工程)、Build(编译)、Tools(工具)、Window(窗口)、Help(帮助)。主窗口的左侧是工作区窗口,右侧是程序编辑窗口,下面是输出窗口和状态栏。工作区窗口显示所建立的工程的信息,程序编辑窗口用来输入和编辑源程序,输出窗口用来显示编译或连接时的错误信息(error)或警告信息(warning)。

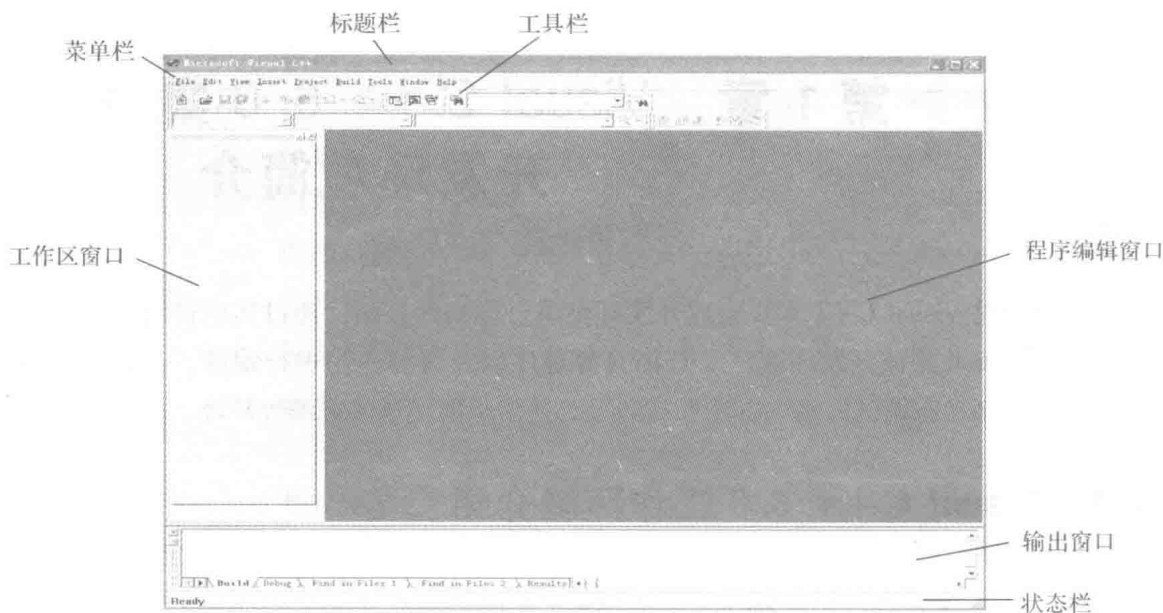


图 1-1 Visual C++ 6.0 的主窗口

1.1.4 Visual C++ 6.0 集成环境的菜单栏

与 Windows 的操作方法相似，Visual C++ 6.0 的菜单选择可以通过两种方法来进行：一是鼠标单击所选的菜单；另一种是键盘操作，即通过相应的热键来选择。选中某个菜单后，就会出现相应的下拉式子菜单。在下拉式子菜单中，有些菜单项的右边对应着相应的快捷键。

(1) File 菜单

File 菜单主要包括一些与文件有关的操作命令，其中的主要菜单项、按钮、快捷键及其功能见表 1-1。

表 1-1 File 菜单中的菜单项和相应功能

菜单项	按钮	快捷键	功能
New		Ctrl + N	创建一个新的文件或者工程
Open		Ctrl + O	打开一个已存在的文件
Close			关闭当前打开的文件
Open Workspace			打开一个已经存在的工作区
Save Workspace			保存当前打开的工作区
Close Workspace			关闭当前工作区
Save		Ctrl + S	保存当前文件
Save As			以新文件名保存当前文件
Save All			保存所有打开的文件

(续)

菜单项	按钮	快捷键	功能
Print		Ctrl + P	打印文件的全部或选定部分
Recent Files			最近的文件列表
Recent Workspace			最近的工作区列表
Exit			退出集成环境

(2) Edit 菜单

Edit 菜单主要包括一些与文件编辑有关的操作命令，它的作用在于为用户提供了一种编辑当前打开文件的手段。Edit 菜单中的主要菜单项、按钮、快捷键及功能见表 1-2。

表 1-2 Edit 菜单中的菜单项和相应功能

菜单项	按钮	快捷键	功能
Undo		Ctrl + Z	撤销上一次编辑操作
Redo		Ctrl + Y	恢复被撤销的编辑操作
Cut		Ctrl + X	剪切选定的内容，并移到剪贴板中
Copy		Ctrl + C	将被选定的内容复制到剪贴板中
Paste		Ctrl + V	将剪贴板中的内容粘贴到当前位置
Delete		Del	删除选定的文本或者光标后面的字符
Select All		Ctrl + A	一次性选定窗口中的全部内容
Find		Ctrl + F	在当前文件中查找指定的字符串
Find in Files			在多个文件中查找指定的字符串
Replace		Ctrl + H	替换指定的字符串
Go To		Ctrl + G	将光标转移到指定的位置
Bookmarks		Alt + F2	给文本文件加标签
Breakpoint		Alt + F9	在指定位置设定断点

(3) View 菜单

View 菜单中的菜单命令项主要用于改变窗口的显示方式和激活指定的窗口。它的作用在于让用户设置窗口的显示方式，并提供观察调试的窗口。View 菜单中的主要菜单项、按钮、快捷键及其功能见表 1-3。

表 1-3 View 菜单中的菜单项和相应功能

菜单项	按钮	快捷键	功能
ClassWizard		Ctrl + W	对工程中的类进行维护
Resource Symbols			浏览、编辑资源文件中的符号常量
Resource Includes			编辑、修改资源符号文件名及预处理命令

(续)

菜单项	按钮	快捷键	功能
Full Screen			在窗口的全屏方式和正常方式之间切换
Workspace		Alt + 0	显示或隐藏工作区窗口
Output		Alt + 2	显示或隐藏输出窗口
Debug Windows			调试程序所需的一些辅助查看命令
Properties		Alt + Enter	编辑当前被选定对象的属性

(4) Insert 菜单

Insert 菜单主要包括一些与添加有关的操作命令，用于向工程中插入类、资源、对象等。Insert 菜单中的主要菜单项、快捷键及其功能见表 1-4。

表 1-4 Insert 菜单中的菜单项和相应功能

菜单项	快捷键	功能
New Class		创建新类并加入到工程中
New Form		创建新窗体并加入到工程中
Resource	Ctrl + R	创建各种资源并加入到工程中

(5) Project 菜单

Project 菜单主要包括一些与工程管理有关的操作命令。Project 菜单中的主要菜单项、快捷键及其功能见表 1-5。

表 1-5 Project 菜单中的菜单项和相应功能

菜单项	快捷键	功能
Add To Project		在工程中增加新文件、文件夹、数据库链接等
Settings		设置当前工程中的各种相关配置
Insert Project into Workspace	Alt + F7	把已经存在的工程插入到当前工作区窗口中

(6) Build 菜单

Build 菜单主要包括一些与建立可执行程序有关的操作命令，可以提供编译、链接、调试等功能。Build 菜单中的主要菜单项、按钮、快捷键及其功能见表 1-6。

表 1-6 Build 菜单中的菜单项和相应功能

菜单项	按钮	快捷键	功能
Compile		Ctrl + F7	编译源代码文件
Build		F7	编译、链接工程，生成可执行文件
Rebuild All			编译和链接所有的文件及资源，生成可执行文件
Batch Build			一次编译和链接多个工程

(续)

菜单项	按钮	快捷键	功能
Start Debug			开始或继续调试程序
Execute	!	Ctrl + F5	运行程序，如果未生成可执行文件，则先编译、链接
Configurations			编辑工程的配置

(7) Tools 菜单

Tools 菜单中的菜单命令主要用于启动 Visual Studio 实用工具和定制 Visual C ++ 集成开发环境界面。Tools 菜单中的主要菜单项及其功能见表 1-7。

表 1-7 Tools 菜单中的菜单项和相应功能

菜单项	功能
Customize	定制集成开发环境的用户界面
Options	改变集成开发环境的各项设置

(8) Window 菜单

Window 菜单主要包括一些与窗口显示有关的操作命令。Window 菜单中的主要菜单项、按钮及其功能见表 1-8。

表 1-8 Window 菜单中的菜单项和相应功能

菜单项	按钮	功能
New Window		为当前文档打开一个新的窗口
Split		分隔窗口
Close		关闭当前打开的窗口
Close All		关闭所有打开的窗口

(9) Help 菜单

Help 菜单包括了有关 Visual C ++ 帮助的操作命令。Help 菜单中的主要菜单项、按钮及其功能见表 1-9。

表 1-9 Help 菜单中的菜单项和相应功能

菜单项	按钮	功能
Contents		启动 MSDN，显示所有帮助信息的内容列表
About Visual C ++	?	显示版本信息

1.2 Visual C ++ 6.0 环境中建立 C 语言源程序

Visual C ++ 6.0 是基于 C/C ++ 的，它可以采用两种方式编写 Win32 应用程序，第

一种是基于 Windows API 的 C 编程方式(又称为控制台编程方式),其代码运行效率较高,但随着工作量的增大开发难度也逐渐增大。第二种是基于 MFC 的 C++ 编程方式,其代码运行效率相对较低,但工作量和开发难度相对较小。本书采用控制台编程方式建立 C 语言源程序。

下面以实例讲解如何利用 Visual C++ 6.0 集成开发环境建立 C 语言源程序并对其进行编译、调试、运行等。为方便实例讲解,首先在 C 盘根目录分别新建两个名为“Cprogram”和“Clanguage”的文件夹以备使用。

【例 1.1】 在 Visual C++ 6.0 集成开发环境中新建一个 C 源程序文件,命名为 hello.c,保存到 C:\Cprogram 文件夹下。

```
#include <stdio.h>
void main()
{
    printf("Hello,everyone. This is your first C program! \n");
}
```

1.2.1 新建 C 语言源程序

例 1.1 中利用 Visual C++ 6.0 建立 C 语言源程序的第一步是新建一个源程序文件。在 Visual C++ 6.0 主窗口中,选择“File”菜单中的“New”命令,在弹出的“New”对话框中单击“Files”选项卡,选择“C++ Source File”选项,在“File”编辑框中填写源文件名,例如“hello.c”,在“Location”编辑框中指定文件所在目录,例如“C:\Cprogram”,如图 1-2 所示,然后单击“OK”按钮完成 C 源文件的新建操作。注意:填写源文件名时一定要加上扩展名“.c”,否则系统会为文件添加默认的 C++ 源文件扩展名“.cpp”。

此时程序编辑窗口成为一个空白的白色区域,在这里就可以输入“hello.c”源程序的内容。输入源程序后,选择“File”菜单中的“Save”命令保存源程序,如图 1-3 所示。

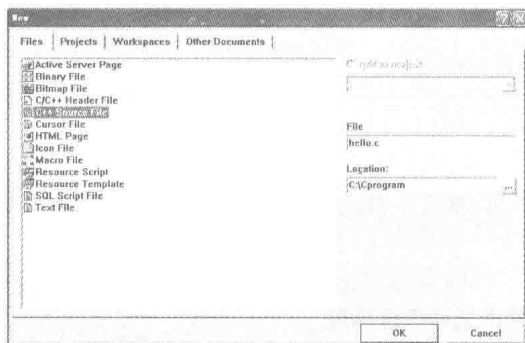


图 1-2 新建一个 C 源程序文件

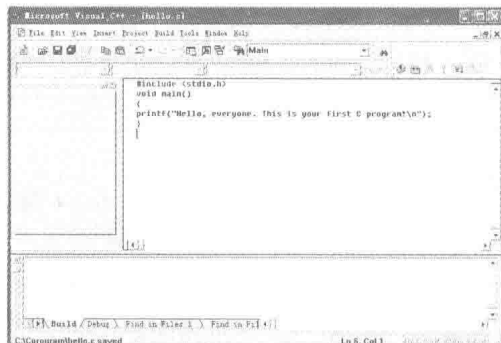


图 1-3 输入 C 源程序

1.2.2 创建工程

利用 Visual C++ 6.0 新建完成源程序文件后,第二步是为该源程序文件创建一个工程(Project),用来存放 C 程序的所有信息。这里介绍创建工程的第一种方法,即利用新建工程中添加一个已有的源程序文件建立工程与源程序间关联的创建方法。

例 1.1 中为一个已有的源程序文件“hello.c”创建一个工程的步骤为:

(1) 打开新建工程界面

在 Visual C++ 6.0 主窗口中, 选择“File”菜单中的“New”命令, 在弹出的“New”对话框中单击“Projects”选项卡, 选择“Win32 Console Application”工程类型, 在“Project name”编辑框中填写工程名, 例如“helloproject”, 在“Location”编辑框中填写工程所在目录, 例如“C:\Cprogram”, 如图 1-4 所示, 然后单击“OK”按钮继续。

(2) 新建一个空白的工程

在弹出的“Win32 Console Application - Step 1 of 1”对话框中, 选择“An empty project”选项, 如图 1-5 所示, 然后单击“Finish”按钮继续。

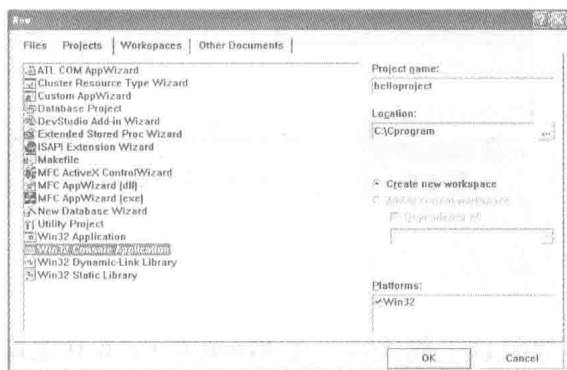


图 1-4 新建工程

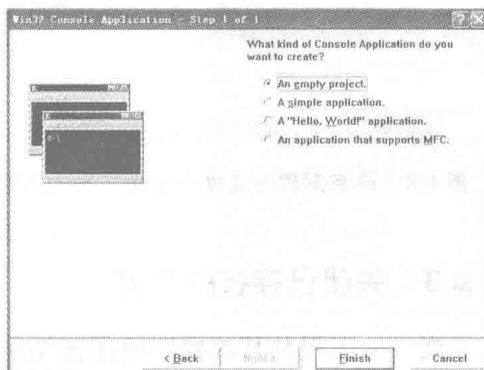


图 1-5 “Win32 Console Application - Step 1 of 1”对话框

(3) 完成工程的创建

在弹出的“New Project Information”对话框中, 单击“OK”按钮完成工程的创建。如图 1-6 所示。此时在“C:\Cprogram”文件夹下自动创建了工作区文件“helloproject.dsw”和工程文件“helloproject.dsp”。

(4) 建立已有源程序文件和新建工程的关联

在 Visual C++ 6.0 主窗口中, 选择“Project”菜单中的“Add To Project”子菜单中“Files”命令, 在弹出的“Insert Files into Project”对话框中, 在默认查找范围“C:\Cprogram”路径下选择“hello.c”文件, 并点击“OK”按钮, 如图 1-7 所示。此时, 在刚才创建好的“helloproject”工程界面的程序编辑窗口中将出现“hello.c”文件内容, 并且将该源程序文件“hello.c”和工程“helloproject.dsp”之间建立了关联, 如图 1-8 所示。

(5) 在工程中显示添加关联后的源程序文件

在当前的“helloproject”工作界面中, 单击左侧工作区窗口下方的“File View”选项

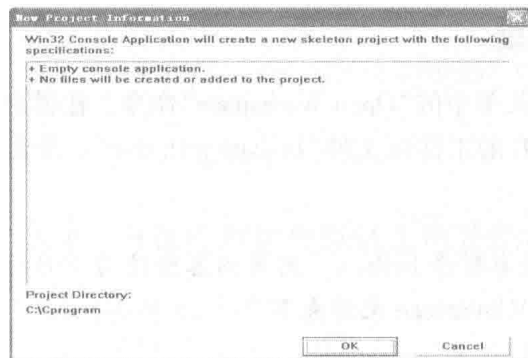


图 1-6 “New Project Information”对话框

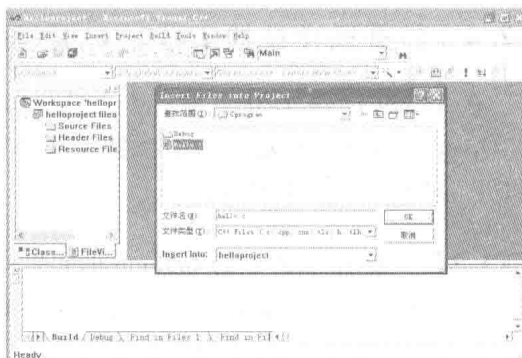


图 1-7 “Insert Files into Project”对话框

卡，以文件视图显示，打开“Source Files”文件夹，双击其中的“hello.c”源程序，即可在程序编辑窗口中显示源程序文件的内容，如图 1-9 所示。至此，例 1.1 中新建一个 C 源程序文件的过程就完成了。

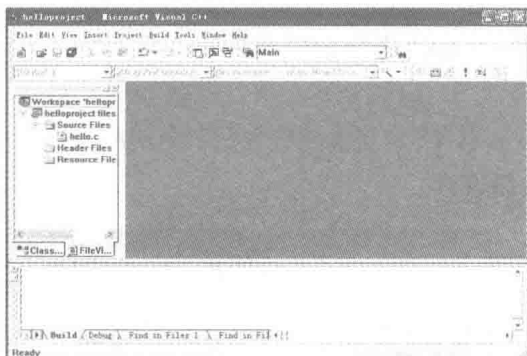


图 1-8 已有源程序文件和新建工程的关联



图 1-9 工程中显示源程序文件内容

1.2.3 关闭已存在的工程

当一个工程及其源程序使用完毕后，需要将该工程关闭。在 Visual C++ 6.0 主窗口中，选择“File”菜单中的“Close Workspace”命令，在弹出的“Microsoft Visual C++”对话框中单击“是”按钮即可关闭该工程，如图 1-10 所示。注意：如果不关闭工程就添加并编写新的源程序，会使原来的源程序还在工作区内，从而给初学者运行程序带来麻烦。

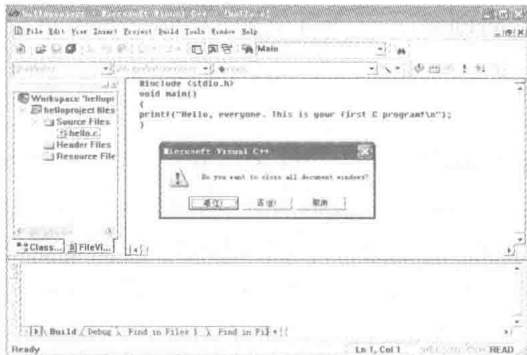


图 1-10 关闭已存在的工程

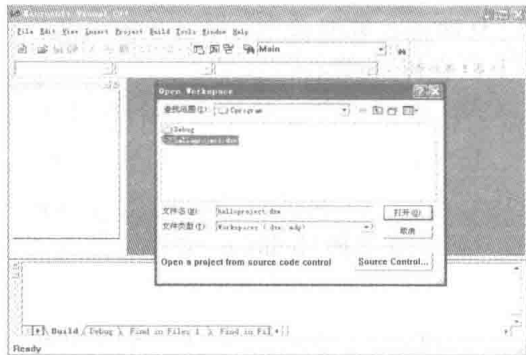


图 1-11 打开已存在的工程

1.2.4 打开已存在的工程对 C 源程序编辑

在 Visual C++ 6.0 主窗口中，选择“File”菜单中的“Open Workspace”命令，在弹出的“Open Workspace”对话框中找到并选择要打开的工作区文件“helloproject.dsw”，单击“打开”按钮，如图 1-11 所示。

【例 1.2】 打开已创建的 helloproject 工程中的源程序 hello.c，将其内容修改为下面的程序，并另存为 welcome.c 文件名，保存到 C:\Clanguage 文件夹下。

```
#include <stdio.h>
void main()
{
    printf("C language is funny! \t Welcome to study C language. \n");
}
```

在已打开的源程序文件“hello.c”中，修改源程序内容后，选择“File”菜单中的“Save As”命令将源程序另存为“welcome.c”文件名，并更改到“C:\Clanguage”文件夹下，如图 1-12 所示。修改保存后的源程序如图 1-13 所示。保存完毕后，需要将该工程关闭。在 Visual C++ 6.0 主窗口中，选择“File”菜单中的“Close Workspace”命令，在弹出的“Microsoft Visual C++”对话框中单击“是”按钮关闭 helloproject 工程。



图 1-12 另存为“welcome.c”文件名



图 1-13 打开已存在工程中的源程序并编辑

1.3 源程序的编译、连接和运行

新建一个 C 源程序文件，为其创建了一个工程并建立了二者的关联之后，就需要对这个源程序进行编译、连接和运行的操作。

【例 1.3】 打开已创建的源程序文件 welcome.c，为其创建默认工程后，将其源程序修改为以下内容后，编译、修改、连接并运行，并在用户屏幕中查看运行结果。

```
#include <stdio.h>
void main()
{
    printf("C language is funny! \t Welcome to study C language. \n")
//缺少分号,语法错误
}
```

1.3.1 为源程序创建默认工程及源程序的编译

(1) 打开已有源程序文件

打开例 1.3 中“C:\Clanguage”文件夹下源程序文件“welcome.c”，在 Visual C++ 6.0 主窗口中，选择“File”菜单中的“Open”命令，在弹出的“打开”对话框中找到并选择要打开的源程序文件“welcome.c”，单击“打开”按钮，如图 1-14 所示。



图 1-14 打开文件对话框

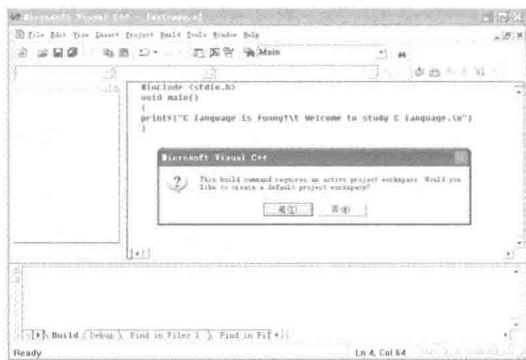


图 1-15 提示创建默认工程对话框

(2) 创建默认工程并对源程序进行编译

将例 1.3 已打开的源程序文件“welcome.c”内容按要求修改保存后，第一步是对源程序文件进行编译操作，得到目标文件“welcome.obj”。

由于例 1.3 中源程序文件“welcome.c”没有创建工程文件，此时需要为其创建一个工程并关联起来，这里介绍第二种创建工程的方法，即为已有源程序创建一个默认的工程。在 Visual C++ 6.0 主窗口中，选择“Build”菜单中的“Compile”命令，或单击工具栏上的  按钮，在弹出的对话框“This build command requires an active project workspace. Would you like to create a default project workspace?”中单击“是”按钮，即可为“welcome.c”源程序创建一个默认的工程，如图 1-15 所示。

创建默认工程的同时，系统将编译当前源程序。下方的输出窗口将显示编译过程中检查出的错误信息(error)或警告信息(warning)。在错误信息或警告信息处双击鼠标左键，可以使输入光标跳转到引起错误或警告的源代码大致位置，以便进行修改，此时光标所在行的左侧会出现一个蓝色箭头。如图 1-16 所示，输出窗口提示“1 error(s), 0 warning(s)”，其中“error C2143: syntax error : missing ';' before '}'”，说明在“}”之前缺少分号，同时在程序编辑窗口一个蓝色箭头标注出错误语句的大致位置。在“printf(“Welcome to study C language. \n”)”语句后面加一个分号后再次编译，输出窗口提示“0 error(s), 0 warning(s)”，如图 1-17 所示。

对于一个正确的源程序，通过“Compile”命令编译成功后可以得到对应的目标文件(.obj)。



图 1-16 源程序编译出错



图 1-17 源程序的编译成功