



语言

程序设计实验教程

刘 强 童 启 主 编



科学出版社

C 语言程序设计实验教程

主 编 刘 强 童 启

副主编 李长云 蒋 鸿 廖立君

杨名念 张宇坤 王志兵

清华大学出版社

北京 100084

http://www.tup.tsinghua.edu.cn

清华大学出版社

北京 100084

http://www.tup.tsinghua.edu.cn

清华大学出版社

北京 100084

http://www.tup.tsinghua.edu.cn

清华大学出版社

北京 100084

http://www.tup.tsinghua.edu.cn

清华大学出版社

北京 100084

http://www.tup.tsinghua.edu.cn

清华大学出版社

北京 100084

http://www.tup.tsinghua.edu.cn

清华大学出版社

北京 100084

http://www.tup.tsinghua.edu.cn

科学出版社

北京 100084

http://www.sciencep.com

内 容 简 介

本书是《C 语言程序设计》(李长云等主编)的配套实验书,目的是帮助学生加深对主教材内容的理解,注重学生程序设计综合能力的培养。本书与理论教材内容相结合,对目前流行的 5 种 C 语言编辑环境进行了介绍,精心设计了 13 个课内实验、12 个课外实验、10 个经典的课程设计及相关的习题与学习指导。本书的实验力争由浅入深、循序渐进地培养学生的编程能力;习题解答则对重难点题目给出了详细的解析,方便学生自学。

本书可以作为高等院校 C 语言程序设计课程的实验教材,还可作为各类计算机培训的教学用书及计算机工作者和爱好者的参考用书。

图书在版编目(CIP)数据

C 语言程序设计实验教程/刘强,童启主编. —北京:科学出版社,2015
ISBN 978-7-03-043210-0

I. ①C… II. ①刘… ②童… 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 021566 号

责任编辑:李淑丽 / 责任校对:李 影
责任印制:霍 兵 / 封面设计:华路天然工作室

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>

文林印务有限公司印刷

科学出版社发行 各地新华书店经销

*

2015 年 3 月第 一 版 开本:787×1092 1/16

2015 年 3 月第一次印刷 印张:16 3/4

字数:397 000

定价:31.40 元

(如有印装质量问题,我社负责调换)

前 言

C 语言是一种通用的程序设计语言,也是普通高等院校常用的一种程序设计教学语言。要学好 C 语言,需要进行大量的实际操作和实践训练。本书通过大量的实验可以帮助学生学习 C 语言程序设计的有关知识,深入理解和掌握 C 语言程序设计所涉及的概念、方法与技巧。

本书是《C 语言程序设计》(李长云、张强主编)的配套实验书,为 C 语言学习者提供上机实验指导、习题解答。全书分为 5 章:第 1 章为 C 语言集成环境简介,介绍了 Visual C++6.0、Code::Blocks、C4Drion、Turbo C++3.0 和 UNIX/Linux 下的 C 语言开发环境;第 2 章为 C 语言程序设计实验,安排了 13 个课内实验,每个课内实验分为“观察与验证”、“分析与改错”、“设计与综合”3 个不同的实验环节,可分层次、循序渐进地进行实验教学;第 3 章为 C 语言程序设计课外实验,供学有余力的同学选做;第 4 章为 C 语言课程设计,提供了 10 个课程设计项目,以培养学生实际分析问题、编程和动手操作的能力,可供不同专业有不同应用要求的学生选做;第 5 章为习题与学习指导,提供主教材配套习题及解答,并对部分重难点题目给出解析,方便学生自学。

本书在内容编排上以验证和观察型实验为基础性实验,以分析型实验培养学生分析问题和解决问题的能力,以设计型和综合型实验训练学生进行程序设计的能力,力图体现因材施教和循序渐进的教学原则,使学生通过实验掌握 C 语言程序设计的基础知识,并提高综合应用的能力。

本书是湖南省普通高等学校省级精品课程“C 语言程序设计”、湖南省普通高等学校特色专业计算机科学与技术的建设与研究成果,本书配套源代码等资源请参见网站 <http://jsjjc.hut.edu.cn>,或联系作者: hutjsj@163.com。

本书由刘强提出编写思路和编写大纲,刘强、李长云、童启、廖立君、蒋鸿、杨名念、王志兵、张宇坤参加编写,最后由刘强统稿。胡泰室同学参与了部分程序的编写与测试,在此表示衷心的感谢。

由于编者水平有限,书中难免有错误和不妥之处,恳请读者批评指正。

编 者

2015 年 2 月

目 录

第 1 章 C 语言集成环境简介	1
1.1 Visual C++集成环境	1
1.2 Code::Blocks	7
1.3 Turbo C++集成环境	17
1.4 UNIX/Linux 下的 C 开发环境	28
1.5 C4droid 手机编程环境	33
第 2 章 C 语言程序设计实验	36
2.1 熟悉 C 语言程序编辑与调试环境	36
2.2 数据类型、运算符、表达式实验	43
2.3 顺序结构程序实验	47
2.4 选择结构程序实验	51
2.5 条件型循环结构程序实验	55
2.6 计数型循环结构程序实验	59
2.7 函数实验	63
2.8 递归、嵌套函数及编译预处理实验	70
2.9 一维数组与二维数组设计实验	74
2.10 字符数组程序实验	80
2.11 指针程序实验	85
2.12 结构体程序实验	90
2.13 文件程序实验	98
第 3 章 C 语言程序设计课外实验	106
3.1 数据类型：简单的数据加密	106
3.2 选择结构：求解不多于 5 位的整数各个数位上的数字	107
3.3 循环结构：爱因斯坦走台阶	109
3.4 循环结构：体育比赛抽签程序	109
3.5 循环与数组综合：十进制转化成二进制实验	110
3.6 数组：学生成绩管理系统	111
3.7 函数：递归程序设计实验	112
3.8 函数：求超级素数	112
3.9 函数：不使用 strcpy 实现两个字符串的比较	113
3.10 结构体：通讯录的建立与查询	114
3.11 文件：英文单词关键词检索	114
3.12 综合程序设计：各类应用题	115
第 4 章 课程设计	116
4.1 课程设计目标	116

4.2 课程设计题目实例	122
第 5 章 习题与学习指导	136
5.1 C 语言程序设计概述	136
5.2 数据类型、运算符和表达式	138
5.3 顺序结构程序设计	143
5.4 选择(分支)结构程序设计	149
5.5 循环结构程序设计	153
5.6 函数	164
5.7 数组	169
5.8 指针	179
5.9 结构体、共用体与枚举	190
5.10 文件	199
5.11 习题解答	204
参考文献	261

第1章 C语言集成环境简介

不同的专业需求、不同操作系统对 C 语言运行环境有不同的要求,目前 Visual C++、Code::Blocks、Turbo C++3.0、Linux 下的 C 环境是最为常用的并能满足不同专业需求的集成环境,而 C4droid 则是智能手机上常用的一种编译器。下面对这几种编程环境进行介绍。

1.1 Visual C++集成环境

Visual C++是 Microsoft 公司 Visual Studio 开发工具箱中的一个 C++程序开发包。Visual Studio 提供了一整套开发 Internet 和 Windows 应用程序的工具,包括 Visual C++、Visual Basic、Visual Foxpro、Visual InterDev、Visual J++及其他辅助工具,包含文本编辑器、资源编辑器、工程编译工具、连接器、源代码浏览器、集成调试工具,以及联机文档。使用 Visual C++可以完成创建、调试、修改应用程序等的各种操作。Visual C++是可视化编程集成工具,可视化技术是当前发展迅速并引人注目的技术之一,它的特点是把原来抽象的数字、表格、功能逻辑等用直观的图形、图像的形式表现出来,可视化编程是它的重要应用之一。所谓可视化编程,就是在软件开发过程中,用直观和具有一定含义的图标按钮、图形化的对象取代原来手工和抽象的编辑、运行、浏览操作,软件开发过程表现为鼠标单击按钮和拖放图形化的对象以及指定对象的属性、行为的过程。这种可视化的编程方法易学易用,而且大大提高了工作效率。Visual C++一般分为三个版本:学习版、专业版和企业版,不同的版本适合于不同类型的应用开发。这里主要是介绍 Visual C++ 6.0 中文专业版。

1.1.1 Visual C++安装与启动

安装 Visual C++ 6.0:运行 Visual Studio 软件中的 Setup.exe 程序,选择安装 Visual C++ 6.0,然后按照安装程序的提示完成安装过程。

启动 Visual C++ 6.0:安装完系统后,可以选择以下两种方式启动。

(1) 点击 Windows “开始”菜单,选择“程序”组下“Microsoft Visual Studio 6.0”子组下的快捷方式 Microsoft Visual C++ 6.0 启动 Visual C++ 6.0 (以下简称 VC++)。

(2) 点击 Windows “开始”菜单,选择“运行”,输入 msdev,即可启动。

如果桌面上有快捷图标,也可双击启动。

1.1.2 Visual C++集成开发环境 (IDE)

集成开发环境 (IDE)是一个将程序编辑器、编译器、调试工具和其他建立应用程序的工具集成在一起的,用于开发应用程序的软件系统。Visual C++软件包中的 Developer Studio 就是一个集成开发环境,它集成了各种开发工具和 VC++编译器。使用者可以在不离开该环境的情况下编辑、编译、调试和运行一个应用程序。IDE 中还提供大量在线帮助信息协助程序员做好开发工作。Developer Studio 中除了程序编辑器、资源编辑器、编译器、调试器外,还有各种工具和向导 (如 AppWizard 和 ClassWizard),以及 MFC 类库,这些都可以帮助程序员快速而正确地开发出应用程序。

Visual C++ 6.0 界面是一个由窗口、工具条、菜单及其他部分组成的一个集成界面,如图 1.1 所示。通过这个界面,用户可以在同一环境下创建、测试、调试应用程序。在开发环境界面中,可以看到在它的上方排列着一系列菜单和工具栏,而每一个菜单下都有各自的菜单命令。在进一步与开发环境打交道之前,我们先了解各个菜单命令的基本功能,因为大部分的操作都是通过菜单来完成的。

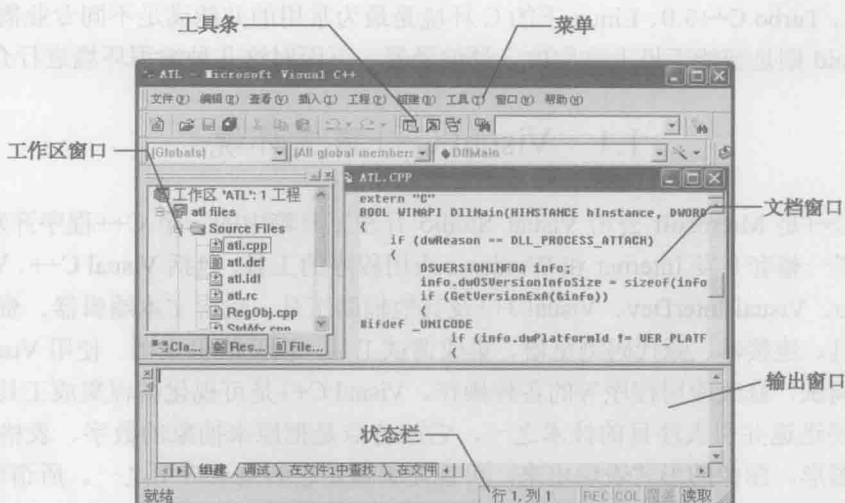


图 1.1 Visual C++集成界面

(1) 工具栏和菜单: 用于提供用户操作的命令接口。菜单以文字和层次化的方式提供命令接口, 工具条由一系列按钮组成。这些按钮是一系列小的位图标志。工具条以图标方式提供快速的命令选择。菜单和工具条在开发的不同进程中有不同显示内容。当第一次打开 Visual C++ 时, 标准的工具条和菜单就会显示出来, 随着开发的不同步骤, 不同的工具条就会自动显示出来, 菜单也会有所变化。工具条有很多种, 你可以显示任意多的工具条, 只要屏幕空间允许。工具条可以任意移动, 也可以放大缩小。工具条和菜单条功能基本相同, 唯一的区别是: 菜单条总占据一行, 并且一般不能隐藏。

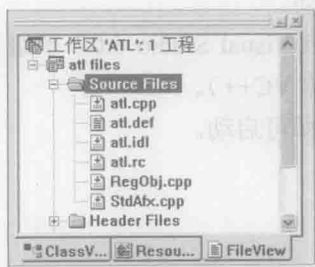


图 1.2 工作区窗口

(2) 工作区窗口: 这个窗口包含关于正在开发的这个项目的有关信息。在没有开发任何项目时, 该窗口显示系统的帮助目录。当打开一个项目以后, 工作区窗口将会显示关于当前项目的文件信息和类的信息。如图 1.2 所示是打开一个项目 ATL 以后的工作区窗口。

(3) 文档窗口区: 这个区域可以显示各种类型的文档, 如源代码文件、头文件、资源文件等, 可以同时打开多个文档。

(4) 输出窗口: 输出窗口用来显示几种信息, 可以通过选择不同的标签显示不同的信息。这些信息包括: 编译连接结果信息 (“组建” 标签)、调试信息 (“调试” 标签)、查找结果信息 (“在文件中查找” 标签)。其中查找结果信息有两个标签, 可以显示两次在文件中查找指定内容的结果。

(5) 状态栏: 状态栏主要显示进行各种操作时的状态信息。

(6) 帮助信息: 大多数时候, 你可以通过按 F1 得到上下文帮助。如在编辑文件时按 F1 可以得到有关编辑的帮助, 在编译连接错误信息上按 F1 可以得到关于该错误的帮助信息。如果想系统地获得帮助, 可以通过选择菜单 “帮助” → “内容” 来启动 MSDN 查阅器, MSDN

查阅器是一个功能强大的程序，可以方便地浏览、查找信息，要想知道具体如何使用 MSDN 查阅器，可以在 MSDN 查阅器中选菜单 Help 下的命令。

(7) Visual C++的编辑器：Visual C++包含一个功能强大的编辑器，可以编辑将被编译成 Windows 程序的 Visual C++源文件。这个编辑器有点像字处理器，但是没有字处理器具备的复杂的排版、文本格式等功能，它注重的是如何帮助程序员快速高效地编制程序。它具有以下特点：

① 自动语法。用高亮度和不同颜色的字来显示不同的语法成分，如注释、关键字和一般代码用不同的颜色显示。

② 自动缩进。帮助你排列源代码，使其可读性更强。

③ 拖放编辑。能够用鼠标选择文本并自由拖动到任意位置。

④ 自动错误定位。能自动将光标移动到有编译错误的源代码处。

⑤ 参数帮助。在编辑时用到预定义的 Windows 函数时，可以自动为你显示函数参数。

⑥ 集成的关键字帮助。能够使你快速得到任何关键字、MFC 类或 Windows 函数的帮助信息（按 F1 即可）。

当打开一个源代码文件时，就可以利用编辑器对其进行编辑。源代码文件在文档显示区显示，每个文件有独立的显示窗口。如果你选择用其他编辑器编辑源文件，必须将它以纯文本的方式保存。VC 的编译器不能处理其中有特别格式字符的文件。

1.1.3 项目和项目工作区

一个 Windows 应用程序通常有许多源代码文件及菜单、工具栏、对话框、图标等资源文件，这些文件都将纳入应用程序的项目中。通过对项目工作区的操作，可以显示、修改、添加、删除这些文件。项目工作区可以管理多个项目。

1. 项目基本概念

在 Windows 环境下，大多数应用程序除了许多源代码文件外，还包含菜单、工具栏、对话框、图标等，Visual C++称它们为资源，这些资源通常用资源文件保存起来。另外，还要包含应用程序代码源文件编连时所需要的库文件、系统 DLL 文件等。有效组织这些文件并维护各源文件之间的依赖关系是应用程序最先要达到的目的，Visual C++中的项目就起这样的作用。实际上，项目作为工作区中的主要内容已加入集成开发环境中，不再需要自己来组织这些文件，只需要在开发环境中进行设置、编译、连接等操作，就可创建可执行的应用程序文件或 DLL 文件。

在 Visual C++中，项目中所有的源文件都是采用文件夹方式进行管理的，它将项目名作为文件夹名，在此文件夹下包含源程序代码文件（.cpp 和.h）、项目文件（.dsp）、项目工作区文件（.dsw），以及项目工作区配置文件（.opt），还有相应的 Debug（调试）或 Release（发行）、Res（资源）等子文件夹。

在开发环境中，Visual C++是通过左边的项目工作区窗口对项目进行各种管理。项目工作区窗口包含三种视图，它们分别是类视图（ClassView）、文件视图（FileView）和资源视图（ResourceView）。

2. 类视图（ClassView）

项目工作区窗口的类视图（ClassView）用以显示项目中的所有各类信息。假设打开的项目名为 ATL，单击项目区窗口底部的 ClassView，则显示出一个标题“atl classes”的树状条目，在它的前面是一个图标和一个套在方框中的符号“+”，单击符号“+”或双击图标，atl 中的所

有类名（包括结构体类型名）将被显示，如图 1.3 所示。

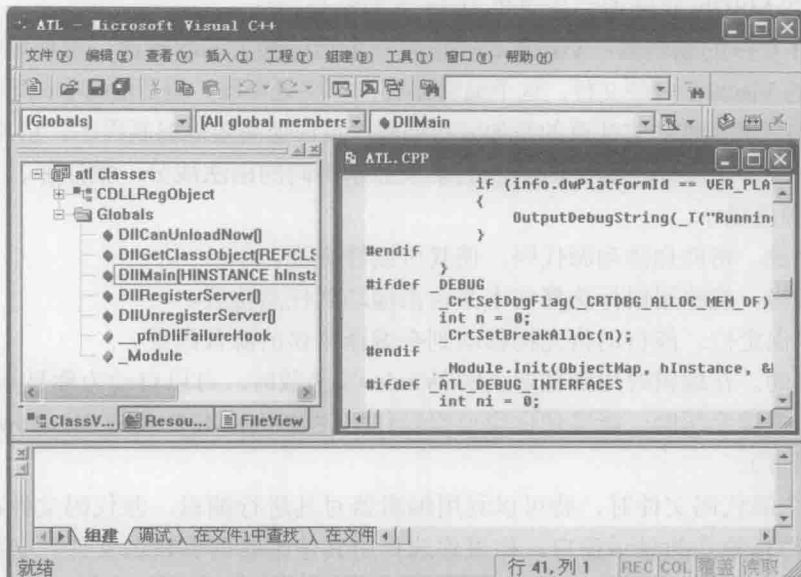


图 1.3 类视图

在类视图（ClassView）中，每个类名前也有一个图标和一个套在方框中的符号“+”，双击图标，则直接打开并显示类定义的头文件（如 exp1.h）；单击符号“+”，则会显示该类中的成员函数和成员变量；双击成员函数前的图标，则在文档窗口中直接打开源文件并显示相应函数体代码。

这里要注意一些图标所表示的含义。例如，在成员函数的图标中，使用紫色方块表示公共成员函数（包括普通函数），使用紫色方块和一把钥匙表示私有成员函数，使用紫色方块和一把锁表示保护型成员函数；又如，用蓝绿色图标表示成员变量等。

3. 文件视图（FileView）

FileView 可将项目中的所有文件（C++源文件、头文件、资源文件、Help 文件等）分类显示，如图 1.4 所示。

每一类文件在文件视图（FileView）中都有自己的目录项，例如所有的 C++源文件都在 Source Files 目录项中。你不仅可以在目录项中移动文件，而且还可以创建新的目录项及将一些特殊类型的文件放在该目录项中。

若创建一个新目录项，可在添加目录项的地方右击，弹出一个快捷菜单，从中选择“New Folder”，将出现如图 1.5 对话框，只要输入目录项名称和相关文件的扩展名，单击“确定”命令按钮即可。

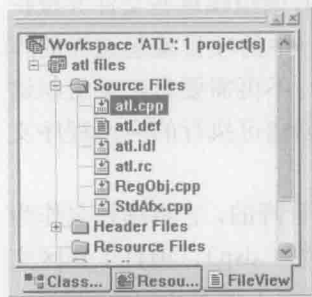


图 1.4 文件视图

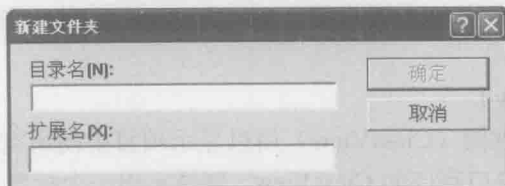


图 1.5 “New Folder”对话框

在大型项目开发中,除了类视图和文件视图外,还包括资源视图(ResourceView)等,后者已超出C程序设计的内容,在这里不作介绍。

1.1.4 项目开发过程

在一个集成的开发环境中开发项目非常容易。一个用VC++开发的项目的通用开发过程如图1.6所示。



图 1.6 项目开发过程

建立一个项目的第一步是利用编辑器建立程序代码文件,包括头文件、代码文件、资源文件等。然后,启动编译程序,编译程序首先调用预处理程序处理程序中的预处理命令(如#include、#define等),经过预处理程序处理的代码将作为编译程序的输入。编译对用户程序进行词法和语法分析,建立目标文件,文件中包括机器代码、连接指令、外部引用以及从该源文件中产生的函数和数据名。此后,连接程序将所有的目标代码和用到的静态连接库的代码连接起来,为所有的外部变量和函数找到其提供地点,最后产生一个可执行文件。一般由一个makefile文件来协调各个部分产生可执行文件。

Visual C++集成开发环境中集成了编辑器、编译器、连接器及调试程序,覆盖了开发应用程序的整个过程,程序员不需要脱离这个开发环境就可以开发出完整的应用程序。

使用Visual C++编辑调试C语言程序的过程将在2.1中讲解,在此不再叙述。

1.1.5 Visual C++常用菜单命令及功能键

VC++开发环境界面由标题栏、菜单栏、工具栏、项目工作区窗口、文档窗口、输出窗口及状态栏等组成。在开发环境界面中,可以看到在它的上方排列着一系列菜单和工具栏,而每一个菜单下都有各自的菜单命令。在进一步与开发环境打交道之前,我们先了解各个菜单命令的基本功能,因为大部分的操作都是通过菜单来完成的。

文件菜单中的命令主要用来对文件和项目进行操作,如“新建”、“打开”、“保存”、“打印”等。

编辑菜单中的命令用来使用户方便快捷地编辑文件内容,如进行删除、复制等操作,其中大多数命令功能与Windows中标准字处理程序的编辑命令一致。

查看菜单中的命令主要用来改变窗口和工具栏的显示方式,激活调试时所用的各个窗口等。

插入菜单中的命令主要用于创建和添加项目及资源。
工程菜单中的命令主要用于项目的一些操作，如向项目中添加源文件等。
工具菜单命令主要用于选择或制订开发环境中的一些实用工具。

窗口菜单中的命令主要用于文档窗口的操作，如排列文档窗口、打开或关闭一个文档窗口、重组或切分文档窗口等。

为了使使用者能够方便快捷地完成程序开发，开发环境提供了大量快捷方式来简化一些常用操作的步骤。键盘操作直接、简单，而且非常方便，表 1.1 是一些最常用的功能键。

表 1.1 File 菜单命令的快捷键及功能描述

菜 单	菜 单 命 令	快 捷 键	功 能 描 述
文件	新建	Ctrl+N	创建一个新项目或文件
	打开	Ctrl+O	打开已有的文件
	文件 保存	Ctrl+S	保存当前文件
编辑	取消	Ctrl+Z	撤销上一次操作
	重做	Ctrl+Y	恢复被撤销的操作
	剪切	Ctrl+X	将当前选定的内容剪切掉，并移至剪贴板中
	复制	Ctrl+C	将当前选定的内容复制到剪贴板中
	粘贴	Ctrl+V	将剪贴板中的内容粘贴到光标当前位置处
	删除	Del	删除当前选定的对象或光标位置处的字符
	断点	Alt+F9	在程序中设置/取消断点
查看	全屏显示		切换到全屏显示方式
	工作空间	Alt+0	显示并激活项目工作区窗口
	输出	Alt+2	显示并激活输出窗口
	调试窗口		操作调试窗口
工程	设置	Alt+F7	修改当前编译和调试项目的一些设置
	导出制作文件		生成当前可编译项目的(.MAK)文件
	插入工程到工作区		将项目加入到项目工作区中
组建	编译	Ctrl+F7	编译 C 或 C++源代码文件
	组建	F7	生成应用程序的 EXE 文件（编译、连接又称编连）
	全部重建		重新编连整个项目文件
	批组建		成批编连多个项目文件
	清除		清除所有编连过程中产生的文件
	开始调试		开始调试，给出调试的一些操作
	执行	Ctrl+F5	执行应用程序
调试	调试→GO	F5	继续执行
	调试→Restart	Ctrl+Shift+F5	重新开始执行
	调试→Stop Debugging	Shift+F5	或 F11（跟踪进函数内）
	调试→Step Into	F11	单步执行（跟踪进函数内）
	调试→Step Over	F10	单步执行（不跟踪进函数）

续表

菜 单	菜单命令	快 捷 键	功 能 描 述
调试	调试→Step Out	Shift+F11	跳出当前函数
	调试→Run to Cursor	Ctrl+F10	执行到光标处
工具	定制		定制菜单及工具栏
	选项		改变开发环境的各种设置
窗口	D 浮动显示	Alt+F6	浮动显示项目工作区窗口
	层叠		层铺所有的文档窗口
	水平平铺		多个文档窗口上下依次排列
	垂直平铺		多个文档窗口左右依次排列
	窗口		文档窗口操作

1.2 Code::Blocks

1.2.1 Code::Blocks 环境简介

C/C++的 IDE (Integrated Development Environment, 集成开发环境) 非常多, 对于学习 C/C++语言的同学而言, 用什么 IDE 可能并不重要, 重要的是学习 C/C++语言本身, 不过, 会用一款自己习惯的 IDE 进行程序的编写和调试确实很方便。

这里主要讲解一款开源、免费、跨平台的集成开发环境 Code::Blocks 的安装、配置, 以及程序的调试和编译等。Code::Blocks 支持十几种常见的编译器, 安装后占用较少的硬盘空间, 个性化特性十分丰富, 功能十分强大, 而且易学易用。我们这里介绍的 Code::Blocks 集成了 C/C++编辑器、编译器和调试器于一体, 使用它可以很方便地编辑、调试和编译 C/C++应用程序。Code::Blocks 具有很多实用的个性化特性, 这里简单介绍几个常用的特性。

1.2.2 Code::Blocks 安装

1. 下载

为了安装 Code::Blocks IDE, 首先需要下载安装程序。我们登陆 Code::Blocks 官网下载 Code::blocks, 网址为 <http://www.codeblocks.org/>。登录后的首页如图 1.7 所示:

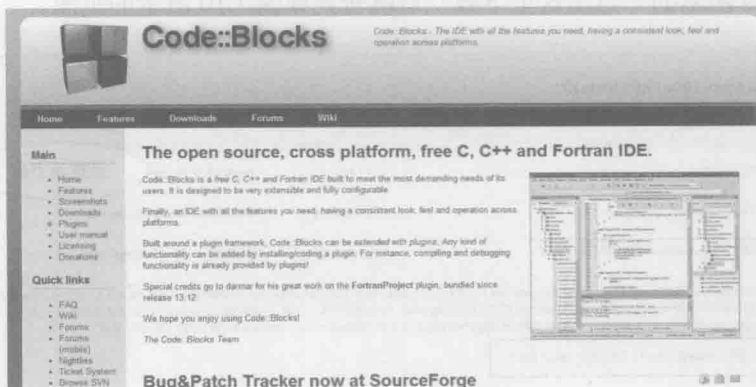


图 1.7 Code::Blocks 官网

建议初学 C/C++ 的同学下载内置 MinGW 的安装程序, 这样不致于花费太多时间配置编译器和调试器, 从而把大部分时间用于学习调试和编写程序。待将来熟悉了 Code::Blocks, 再搭配 GW 或者其他编译器一起使用。点击如图 1.8 所示的“点这里下载”后跳转到如图 1.9 所示的页面。



图 1.8 Code::Blocks 官网下载 (1)

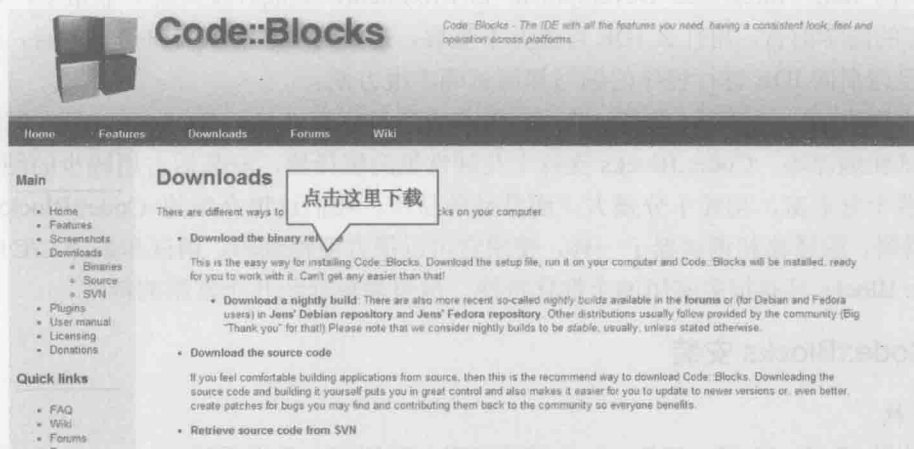


图 1.9 Code::Blocks 官网下载 (2)

点击如图 1.9 所示的“点这里下载”后跳转到如图 1.10 所示的页面。

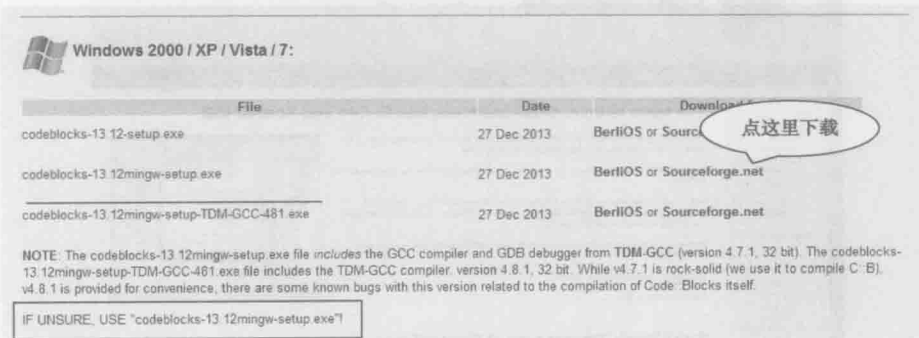


图 1.10 Code::Blocks 官网下载 (3)

点击如图 1.10 所示的“点这里下载”后将转到下载页面，选择保存的路径，即下载完毕。

2. 安装

安装过程比较简单，只需要运行下载后的安装文件，按照安装程序窗口的提示一步步点击，即可安装完成。

1.2.3 Code::Blocks 环境配置

1. 启动

第一次启动 Code::Blocks，可能会出现如图 1.11 所示的对话框，自动检测到 GNU GCC Compiler 编译器，用鼠标选择对话框右侧的 Set as default 按钮，然后再选择 OK 按钮即可。

假如 Code::Blocks 安装正确的话，接下来就进入 Code::Blocks 的主界面，但是会弹出一个标签为 Tips of the Day 的小对话框，如图 1.12。把 Show tips at startup 前面的勾去掉，然后选择 Close，这样下次启动就不会再出现这个小对话框了。

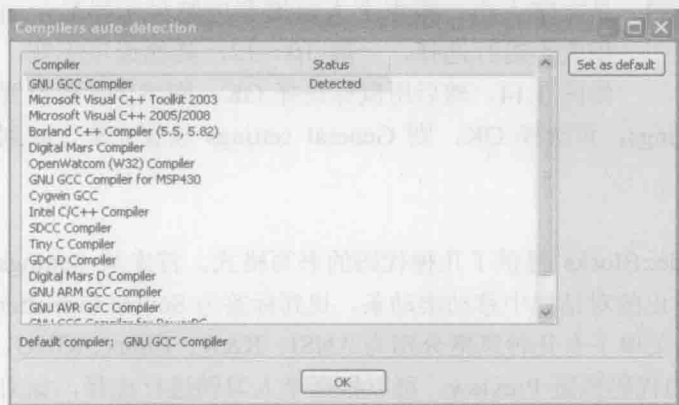


图 1.11 初次启动界面

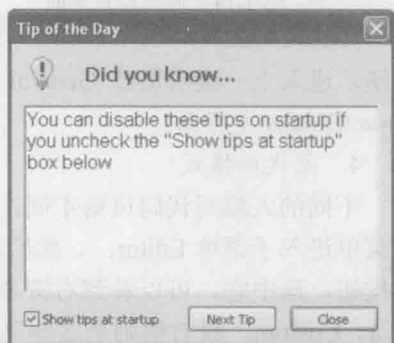


图 1.12 启动提示窗口

启动后的窗口如图 1.13 所示，即为 Code::Blocks 的工作界面。

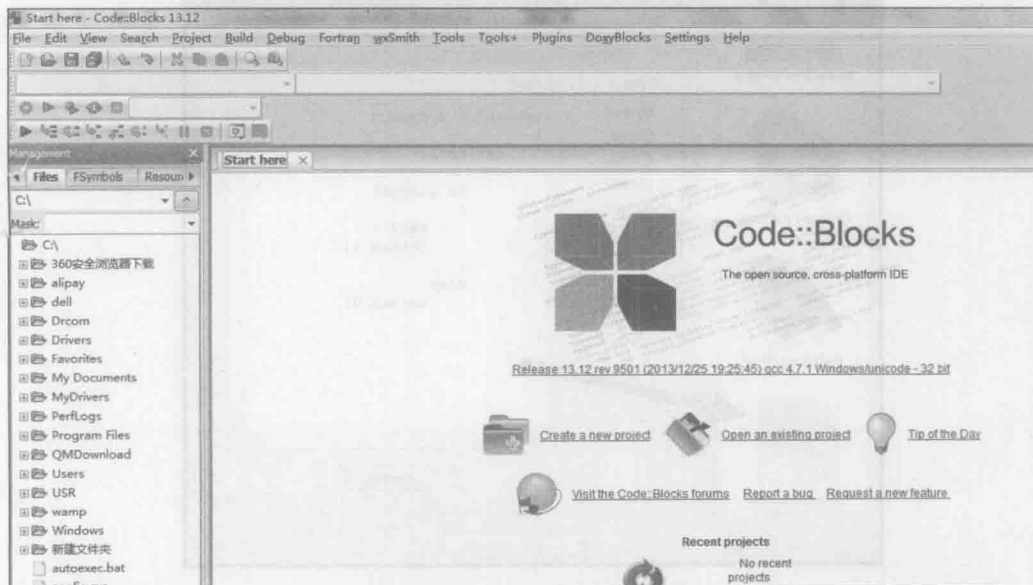


图 1.13 为 Code::Blocks 的工作界面

2. 编辑器

编辑器主要用来编辑程序的源代码, Code::Blocks 内嵌的编辑器界面友好, 功能比较完备, 操作也很简单。

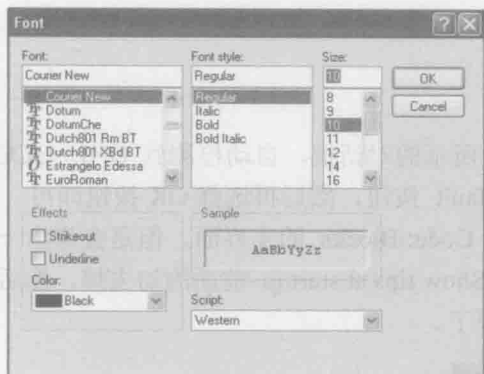


图 1.14 通用设置界面

3. 通用设置

启动 Code::Blocks, 选择主菜单 Settings 下的子菜单 Editor... 会弹出一个对话框, 默认通用设置 General settings 栏目, 选中一些选项如图 1.14 所示。

然后设置字体, 字体设置首先选择右上角的 Choose 按钮, 会弹出一个对话框, 对话框主要有三个竖向栏目, 最左侧的栏目 Font 用来选择字体类型, 建议选择 Courier New, 中间栏目 Font style 是字体样式, 建议选择 Regular, 最右边的栏目 Size 是文字大小, 根据个人习惯和电脑显示器显示面积大小进行选择, 一般 10~12, 其他选项不变, 如图 1.14。然后用鼠标选择 OK, 则字体参数设置

完毕, 进入上一级对话框 General settings, 再选择 OK, 则 General settings 设置完毕, 回到 Code::Blocks 主界面。

4. 源代码格式

不同的人编写代码风格不同, Code::Blocks 提供了几种代码的书写格式。首先从 Settings 主菜单进入子菜单 Editor..., 然后从弹出的对话框中移动滚动条, 找到标签为 Source formatter 的按钮, 选中它, 可以看到右侧 Style 菜单下有几中风格分别为 ANSI、K&R、Linux、GNU、Java、Custom, 最右侧则是这些风格的代码预览 Preview。可以根据个人习惯进行选择, 如果选择 Custom 则需要自己设置两个子菜单 Indentation 和 Formatting 下的各个选项, 选中自己习惯或者喜欢的风格 (笔者的习惯是用 ANSI), 然后点击 OK 按钮即可, 如图 1.15 所示。

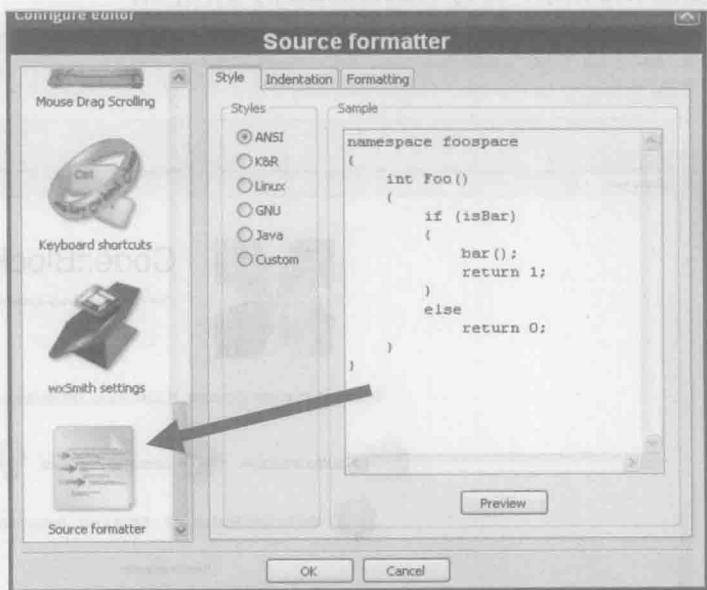


图 1.15 源代码格式设置界面

如此以来，编辑器的基本设置就完成了，尽管还有很多其他的选项和参数，但是并不太常用，因此这里就不做详细介绍了。

1.2.4 程序编写与运行

1. 建立工程

下载安装 Code::Blocks 并配置完成以后，就可以编辑调试程序了。利用 Code::Blocks 创建一个工作空间（workspace）跟踪你当前的工程（project）。如果有必要，还可以在当前空间创建多个工程。一个工程就是一个或者多个源文件（包括头文件）的集合。源文件（source file）就是程序中包含源代码的文件，如果正在编写 C 程序，也就是在编写 C 源代码（文件后缀名为.c）。创建库文件（library files，文件后缀名为.h 或.hpp）时，会用到头文件（header file）。一个库（library）是为了实现特定目标的函数集合，例如数学运算。

创建一个工程可以方便地把相关文件组织在一起。一个工程刚建立时，一般仅仅包含一个源文件。但是，伴随着编程经验的增长可能会用到更复杂的工程，此时一个工程可能包含很多源文件和头文件。

创建一个工程如图 1.16 所示。

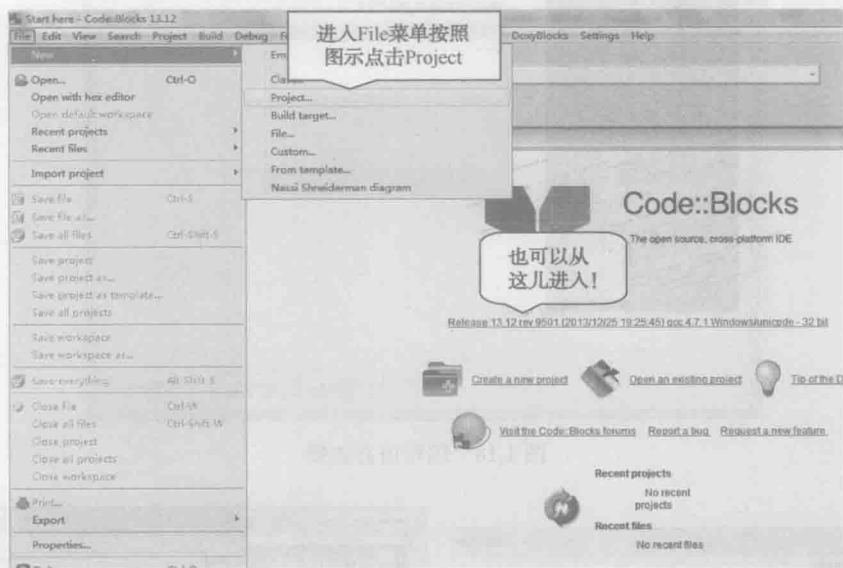


图 1.16 建立工程界面（1）

点击后出现如图 1.17 所示的对话框。这个窗口中含有很多带有标签的图标，代表不同种类的工程。最常用的是 Console application，用来编写控制台应用程序，其他的是一些更高级的应用。

选择 Console application（控制台应用程序）。点击 Go，接着点击 Next 后出现如图 1.18 所示的对话框，要求选择编程语言。由于是学习 C 语言，我们选择“C”。

编程环境选择完毕之后，接着进入项目设置界面，如图 1.19 所示，依次输入项目名称、项目路径，并选择默认的编译器，如图 1.20 所示，即可进入程序编辑界面，如图 1.21 所示。