

第 5 章

CodeBlocks 集成开发环境介绍

CodeBlocks 是一个开放源码的全功能的跨平台 C/C++ 集成开发环境,它支持十几种常见的编译器,个性化特性非常丰富,功能强大,易学易用。

CodeBlocks 由纯粹的 C++ 语言开发完成,它使用了著名的图形界面库 wxWidgets(3.x) 版,集成了 C/C++ 编辑器、编译器和调试器于一体,能方便地编辑、调试和编译程序。对于追求完美的 C++ 程序员,CodeBlocks 功能强大、速度快、完全免费,因此在推出后很快得到了广大程序员的响应。

CodeBlocks 同时提供了分别适应 Windows XP/Vista/7/8.x/10、Linux 32 and 64-bit、macOS X 操作系统的 3 种版本,用户根据自己的操作系统选择相应版本下载即可。建议初学者使用自带编译器的版本,否则手动配置编译器较为麻烦,所以下载时选择带 mingw 的安装包下载。目前最新的适应 Windows 平台的版本为 20.03。

5.1 启动 CodeBlocks

CodeBlocks 是一个 C/C++ 集成开发环境,可以用来实现 C/C++ 程序的编辑、预处理/编译/连接、运行和调试等。启动 CodeBlocks 有以下两种方法。

(1) 单击任务栏中的“开始”按钮,选择“所有程序”→CodeBlocks 命令(Windows 10 系统中,刚安装的软件在菜单项下方有“最近添加”字样),打开子菜单,如图 5.1 所示。单击 CodeBlocks 菜单项,即可启动 CodeBlocks 集成开发工具。

(2) 直接双击安装程序生成在桌面上的 CodeBlocks 图标,如图 5.2 所示,也可以启动 CodeBlocks。

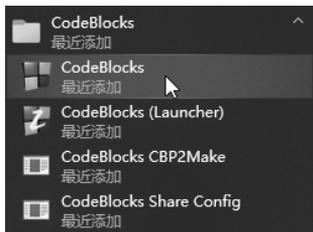


图 5.1 CodeBlocks 子菜单



图 5.2 CodeBlocks 图标

采用以上两种方法均能启动 CodeBlocks, 打开的主窗口如图 5.3 所示。

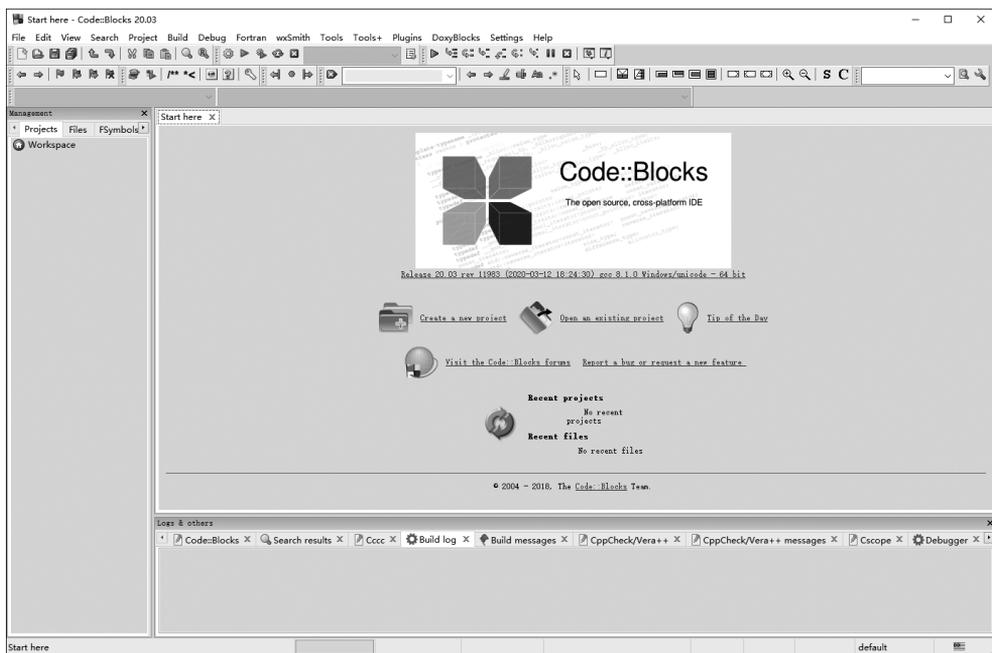


图 5.3 CodeBlocks 主窗口

在主窗口的顶部是 CodeBlocks 的菜单栏。其中包含 15 个菜单项,常用的有 File(文件)、Edit(编辑)、View(查看)、Search(搜寻)、Project(项目)、Build(构建)、Debug(调试)、Settings(设置)和 Help(帮助)等。

主窗口左侧是项目工作管理区,用来显示所设定工作区的信息和所有子程序。右侧是程序编辑窗口,用来输入和编辑源程序。下方是信息显示窗口,主要查看编译信息等。

5.2 新建源程序

新建 C 语言源程序文件的方法如下。

(1) 在主窗口中,依次选择 File→New→File 命令,如图 5.4 所示。

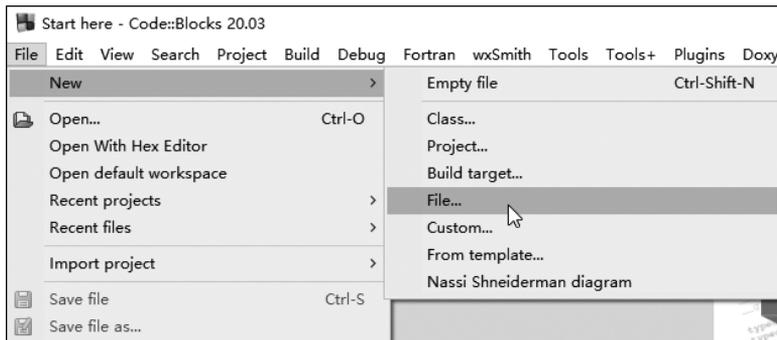


图 5.4 新建源程序文件

(2) 打开 New from template 对话框,如图 5.5 所示。

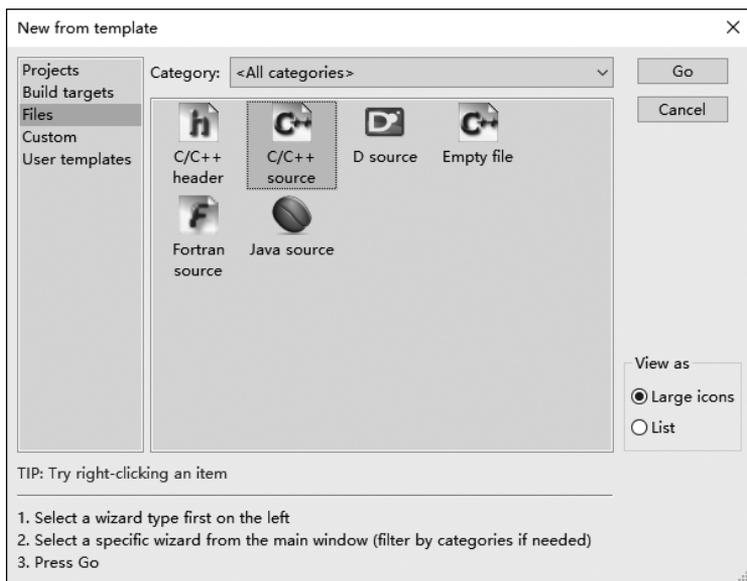


图 5.5 New from template 对话框

(3) 选择 C/C++ source,并单击 Go 按钮,打开欢迎向导对话框,如图 5.6 所示。在该对话框中,若选中 Skip this page next time 复选框,下次将不再显示该页面。

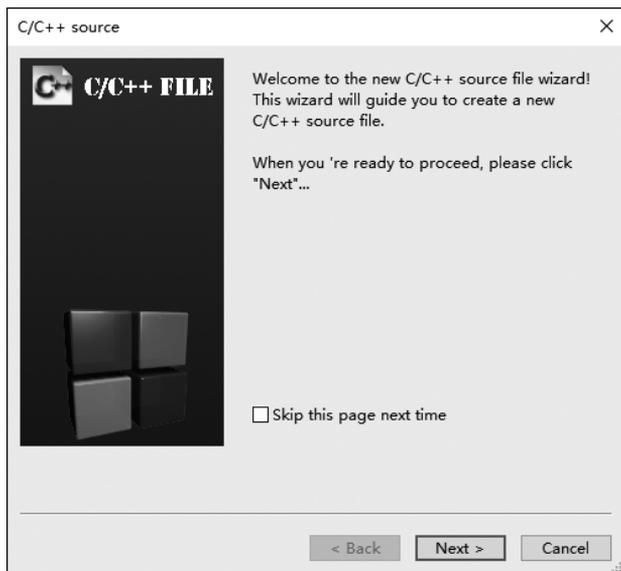


图 5.6 欢迎向导对话框

(4) 单击 Next 按钮,打开语言选择对话框,如图 5.7 所示。

(5) 默认选择是 C++ 语言,即编写 C++ 程序。如果编写 C 语言程序,则选择 C,这里选择 C 后,单击 Next 按钮,打开如图 5.8 所示的设置路径和文件名对话框。

(6) 此处需要设置文件的完整路径(文件的保存位置)及文件名,可直接输入,或单击后

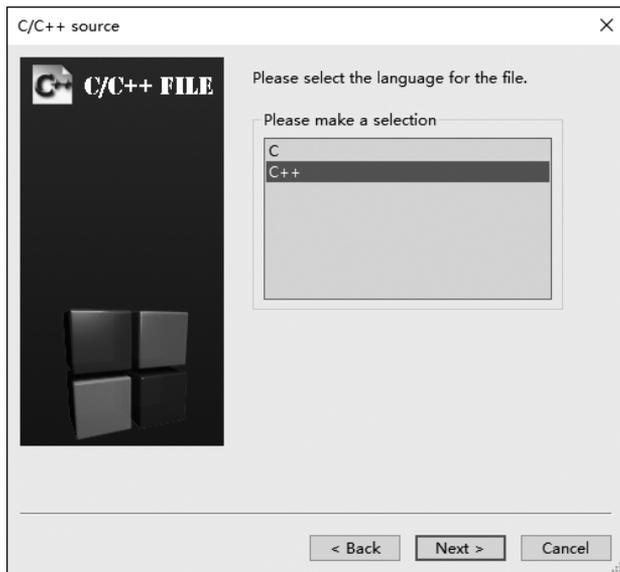


图 5.7 语言选择对话框

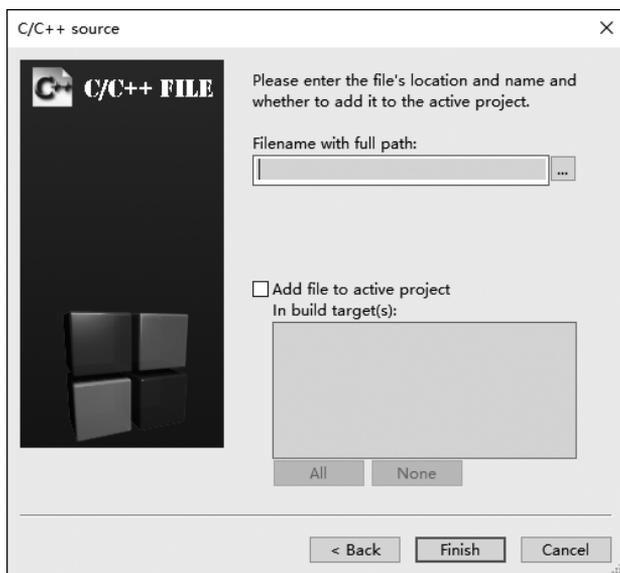


图 5.8 设置路径和文件名对话框

面的“浏览”按钮,打开 Select filename 对话框,如图 5.9 所示,此处把保存位置设为 D:\CTest,“文件名”编辑框中可输入 ujn1004。注意,保存文件的路径尽量不要包含中文,或者说保存文件的各级文件夹名字尽量不要以中文命名,因为在有的系统中偶尔会出现编译问题。

说明: 文件命名时扩展名必须是.c 或.cpp,.c 表示是 C 语言源程序文件,.cpp 表示是 C++ 语言源程序文件。

(7) 单击“保存”按钮后,返回设置路径和文件名对话框,可以看到系统自动在 Filename

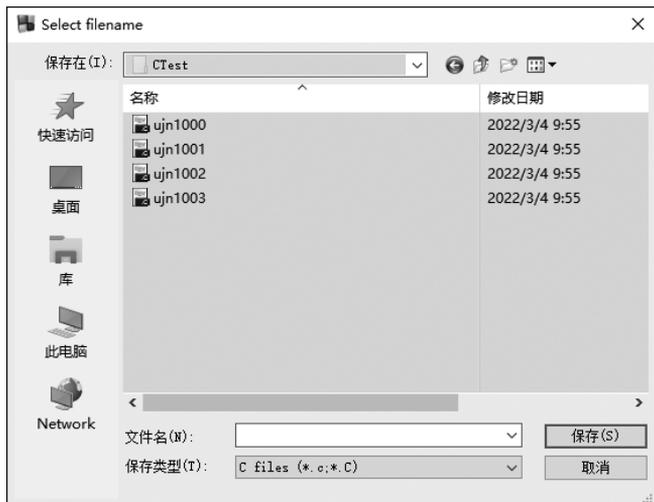


图 5.9 Select filename 对话框

with full path 编辑框中填入了 D:\CTest\ujn1004.c。如果熟悉路径,也可以直接手动输入。设置完成后,单击 Finish 按钮,进入编辑模式,光标在主窗口编辑区第 1 行跳动,同时左侧显示行号,然后就可以输入和编辑源程序了。

上述建立新文件的方法稍微复杂,下面介绍一种较为快速的方法。

① 在 CodeBlocks 主窗口中,依次选择 File→New→Empty file 命令,或按快捷键 Ctrl+Shift+N,新建一个默认名称为 Untitled* 的文件,其中 * 为数字 1、2、3、…。

② 保存文件,给文件命名,如 ujn1005.c,然后就可以输入和编辑源程序了。

5.3 保存源程序

保存源程序方法如下。

(1) 对于已命名过的源文件,保存方法是在 CodeBlocks 主窗口中,依次选择 File→Save file 命令,如图 5.10 所示,或直接按快捷键 Ctrl+S。

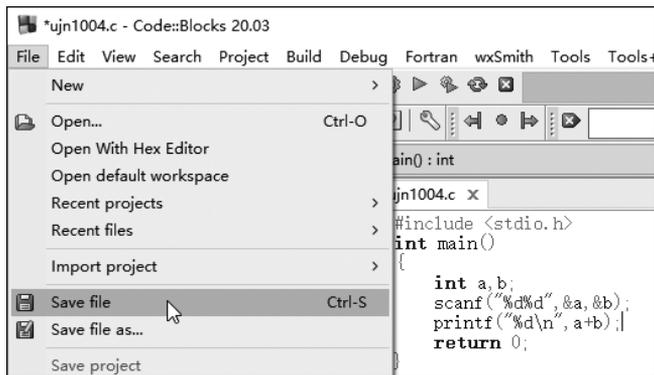


图 5.10 保存文件

说明：未命名过的文件第一次保存只能选择 Save file as 命令,有名字的文件就可以选择 Save file 命令,或用快捷键 Ctrl+S 快速保存。

(2) 对于未命名过的源文件,保存方法是在 CodeBlocks 主窗口中,依次选择 File→Save file as 或 Save file 命令,或者直接按快捷键 Ctrl+S,均会弹出 Save file 对话框,如图 5.11 所示。

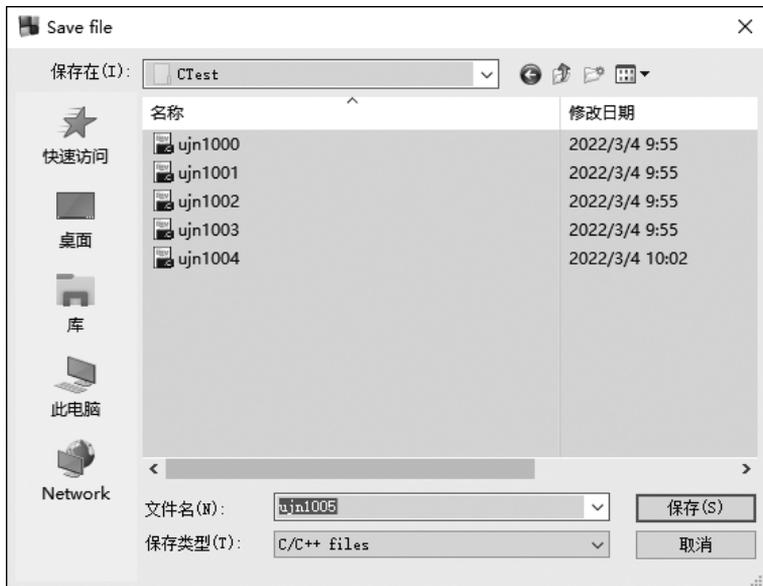


图 5.11 Save file 对话框

在左侧列表中指定文件的保存位置,路径会显示在上方的列表中(此处指定为 D:\CTest),在“文件名”列表中输入文件名称(此处为 ujn1005),在“保存类型”列表中选择保存类型。需要注意的是,在“保存类型”处一定要选择 C/C++ files,意为保存的是一个 C/C++ 语言源文件,默认扩展名为.c。然后单击“保存”按钮,在 CTest 目录下保存为名为 ujn1005.c 的源文件。

5.4 编辑源程序

完成以上操作,即可在编辑区输入程序代码。在输入源代码的过程中,记得要随时对程序进行保存(使用菜单 File→Save file,或直接按快捷键 Ctrl+S),此时会将程序保存到已命名的文件中。如果想将程序保存到其他路径下,可执行 Save file as 命令,指定文件的名称和保存路径。编辑完后的 ujn1005.c 程序代码如图 5.12 所示。

说明：对于未保存的源文件,在编辑区上方的文件名前有星号“*”,表示程序有过更改,还没有保存,保存后该标志消失。若觉得编辑区的字号小或大了,可按住 Ctrl 键,再滚动鼠标滚轮,调整字号大小。

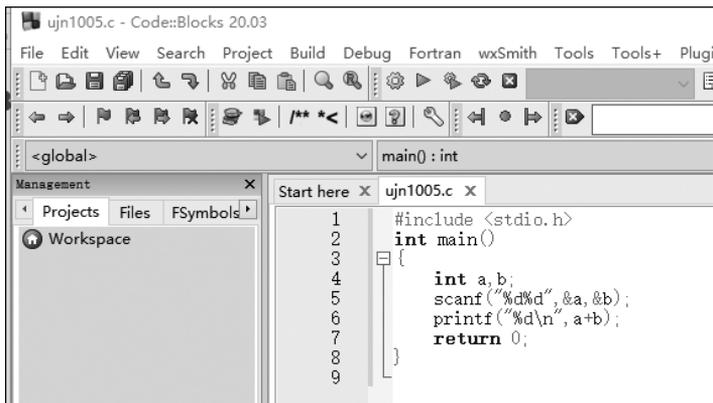


图 5.12 编辑源程序

5.5 编译与连接程序

程序编辑完成后,就可以编译和运行程序。单击菜单选择 Build→Build 命令,或直接按快捷键 Ctrl+F9,可以一次性完成程序的预处理、编译和连接过程。如果程序中存在词法、语法等错误,则编译过程失败,编译器将会在屏幕右下角的 Build messages 标签页中显示错误信息,如图 5.13 所示,并且将源程序相应的错误行号处标记成红色方块。

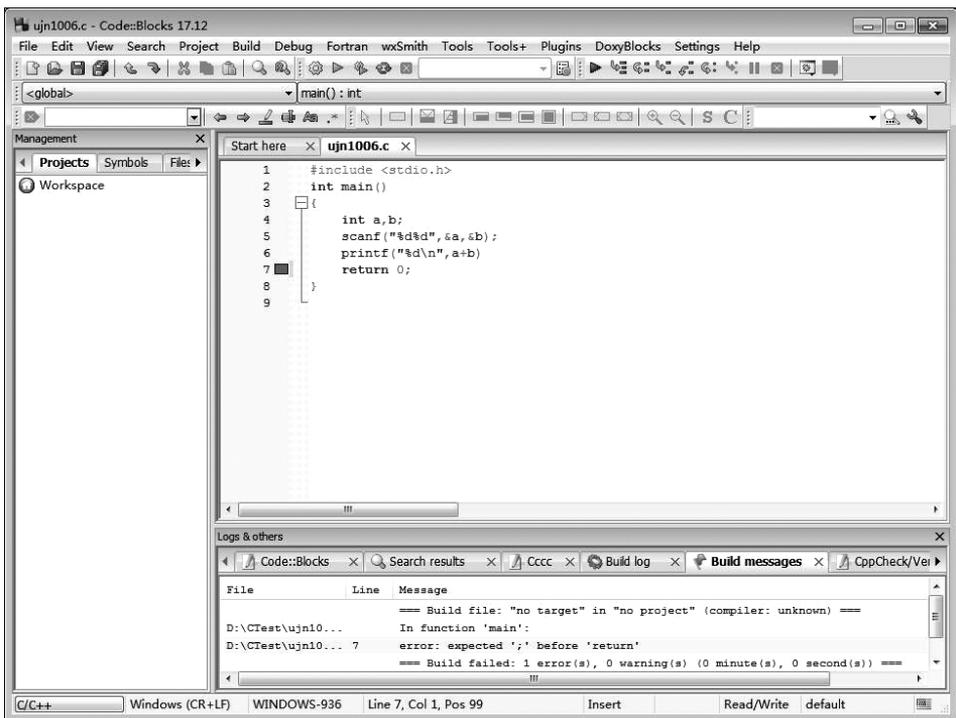


图 5.13 编译错误提示

Build messages 标签页中显示的错误信息是寻找错误原因的重要信息来源,要学会看这些错误信息,在每一次碰到错误且最终解决错误时,要记录错误信息以及相应的解决方法。以后看到类似的错误提示信息时,能熟练反应出是哪里有问题,从而提高程序调试效率。

如果修改了程序中全部的词法、语法等错误后,再次编译,将在“编译日志”标签页中显示编译成功,显示: 0 error(s), 0 warning(s),如图 5.14 所示。此时,在源文件所在目录下将会生成一个同名的.exe 可执行文件(如 ujn1005.exe)。

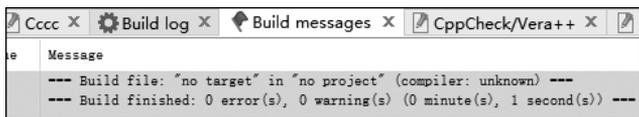


图 5.14 编译成功

说明: 如果执行 Build 命令后,出现如图 5.15 所示的 Environment error 提示,或者执行运行后,出现如图 5.16 所示的提示信息,这是由于编译环境路径设置不对,需按照下面的方法重新设置。



图 5.15 Environment error 提示

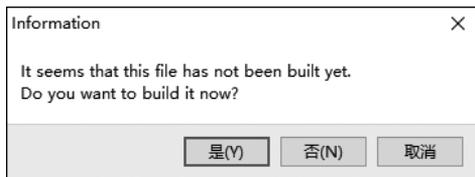


图 5.16 提示信息

① 依次选择菜单栏 Settings→Compiler,打开 Compiler settings 对话框,如图 5.17 所示。

② 在左侧的选项卡中,默认打开的是 Global compiler settings 选项卡,然后在右侧选择 Toolchain executables 标签页,如图 5.18 所示。其中,Compiler's installation directory 处默认被设置为 C:\MinGW 或其他路径,因为系统在这个文件夹下找不到编译所需文件,或者这个文件夹根本就不存在。

③ 单击右侧的 Auto-detect 按钮,系统自动检测,弹出如图 5.19 所示的对话框。

④ 系统检测到在安装路径 C:\Program Files\CodeBlocks\MinGW 下有所需的文件,单击“确定”按钮后,系统将该路径填入图 5.18 中的 Compiler's installation directory 文本框中。

⑤ 在 Compiler settings 对话框中,单击 OK 按钮,完成路径设置。设置完成后,再重新编译或运行程序即可。

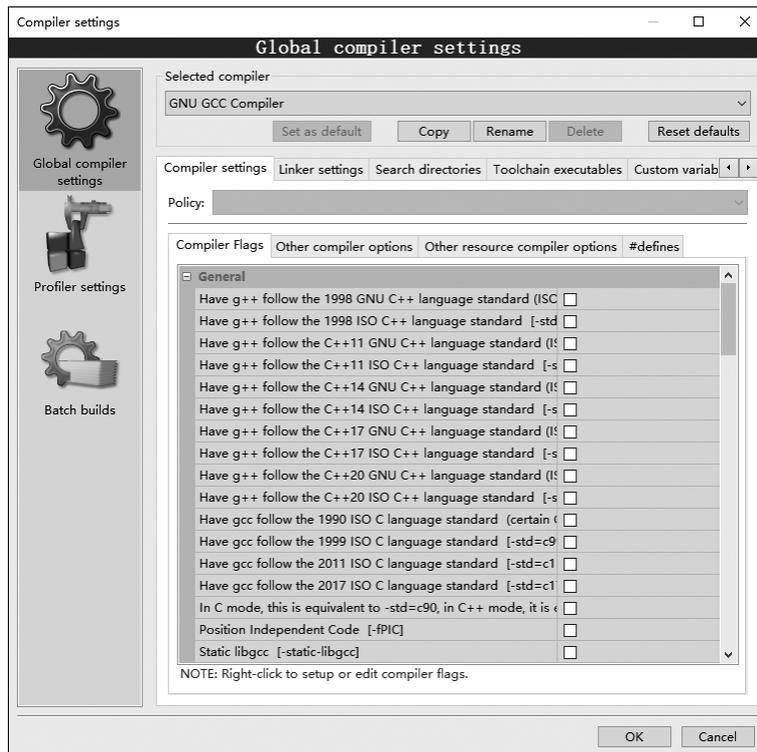


图 5.17 Compiler settings 对话框

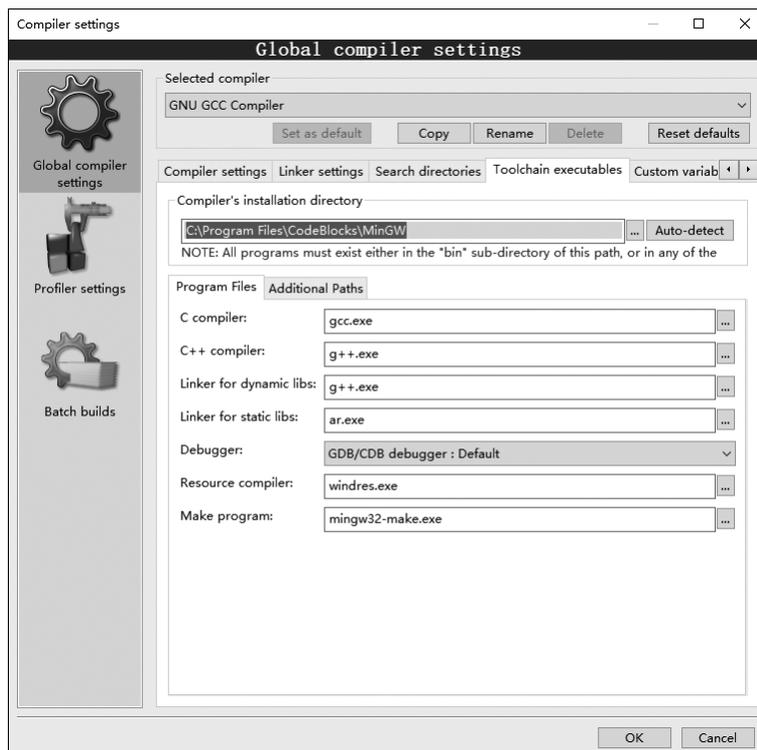


图 5.18 选择 Toolchain executables



图 5.19 系统自动检测对话框

5.6 运行程序

对程序进行编译和连接后,有两种方法可以运行程序。

(1) 双击生成的.exe 文件。

(2) 在 CodeBlocks 环境下,单击菜单选择 Build→Run 命令,或者按快捷键 Ctrl+F10 运行程序。

如运行程序 ujn1006.c,按要求输入数据后,在窗口中显示运行结果、main 函数的返回值及程序运行时间,如图 5.20 所示。

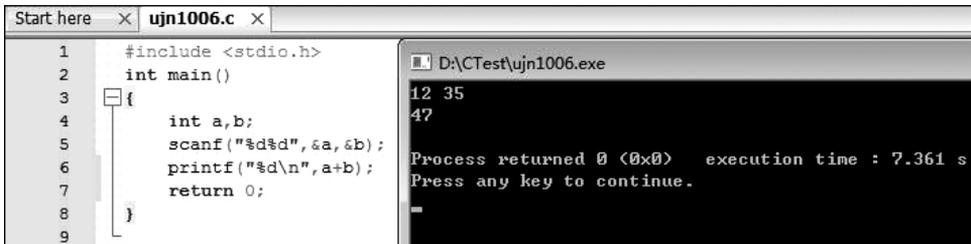


图 5.20 运行程序

说明: 在工具栏上,有编译和运行的快捷按钮,各图标的含义如图 5.21 所示。可直接单击快捷按钮编译和运行程序。执行 Build→Build and run 命令,或按快捷键 F9,或单击快捷按钮,一步完成编译和运行。

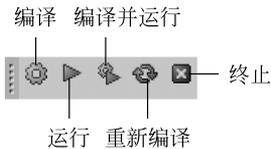


图 5.21 快捷按钮

5.7 调试程序

通过编译和连接的程序仅说明程序中没有词法和语法等错误,而无法发现程序深层次的问题(如算法不对导致结果不正确)。当程序运行出错时,需要找出错误原因。仔细阅读程序来寻找错误固然是一种方法,但是有时光靠读程序已经解决不了问题,此时需要借助 IDE