

Dev C++简明使用手册【摘录】(2008-08-13 09:28:16)

标签: 电脑 dev-c 主菜单 断点 杂谈

这篇文章是我在网络上无意中发现的,很早之前写的,现在的 DEV C++ 早已发展了新版本,但是对于大多数人而言,这篇文章足以够用。使用 DEV C++ 的好处在于,集成使用了 GNU 的 GCC,对于 ANSI C/C++ 的兼容性远好于微软的 VS 系列。相信对于参加 ACM 等需要用 GCC 处理代码的同学而言是不错的选择。原文如下:

Dev-C++ 是一个可视化集成开发环境,可以用此软件实现 C/C++ 程序的编辑、预处理/编译/链接、运行和调试。本手册中介绍了 Dev-C++ 常用的一些基本操作,每一位同学都要掌握。

一. 启动 Dev-C++

方法一:

1. 鼠标点击任务栏中的“开始”按钮,选“程序”菜单项,然后选“程序”下的子菜单项“Bloodshed Dev-C++”项,显示该项下的子菜单。
2. 单击“Dev-C++”菜单项,即可启动 Dev-C++ 集成开发工具。(如图 1 所示)



图 1

方法二:

直接单击桌面上的 Dev-C++ 的图标。

二. 新建源程序

1. 从主菜单选择“File”→“New”→“Source File” (“文件”→“新建”→“源代码”)即可 (如下图 2 所示)。

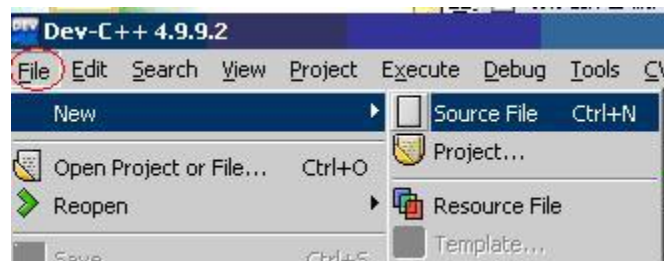


图 2

如果大家看到界面上的字是中文的，则可以根据以下操作将界面改为英文。

点击主菜单“工具”->“环境选项”，在弹出的对话框中选择界面页，在 Language 下拉列表中选择 Chinese 即可，如下图所示。此时界面上的菜单、工具条等全部以英文命名。



2. 此时屏幕右下侧出现一片白色区域，可以在此输入程序。如下图 3 所示。

输出标签页

源程序编辑区域

工具栏
菜单

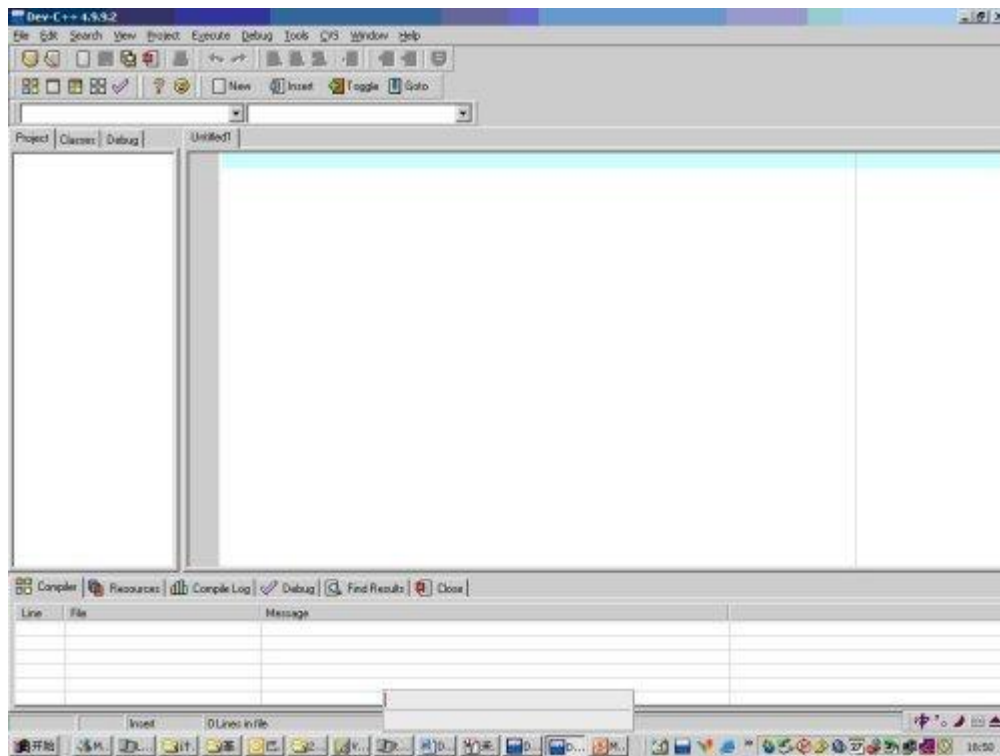


图 3

3. 保存源程序到硬盘

一个好的习惯是创建了一个新程序后，在还未输入代码之前先将该程序保存到硬盘某个目录下，然后在程序的编辑过程中经常性地保存程序，以防止机器突然断电或者死机。要保存程序，只需从主菜单选择“File”→“Save”就可以将文件保存到指定的硬盘目录。如图 4 所示。

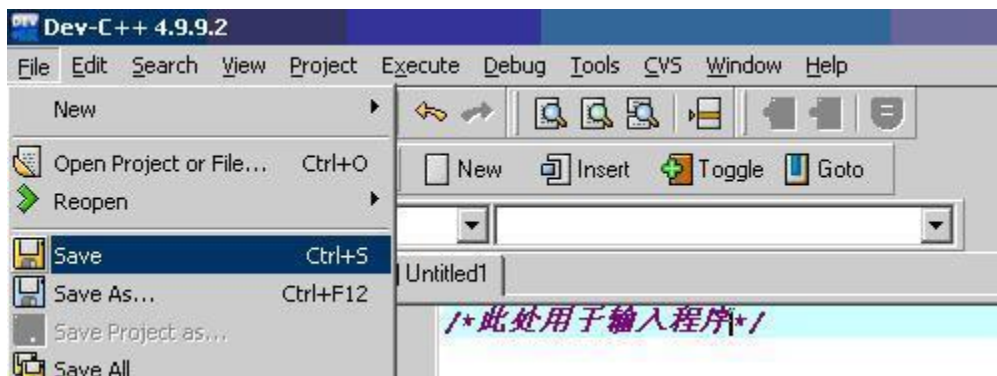


图 4

此时会弹出一个对话框，如图 5 所示。在此你需要指定文件要存放的目录（此处为 F:\temp），文件名称(此处为 test)以及保存类型。需要注意的是，在保存类型处一定要选择 C source files(*.c)，意思是保存地是一个 C 文件。在点击右下角的保存按钮后，在 temp 目录下将会出现一个名为 test.c 的源文件。

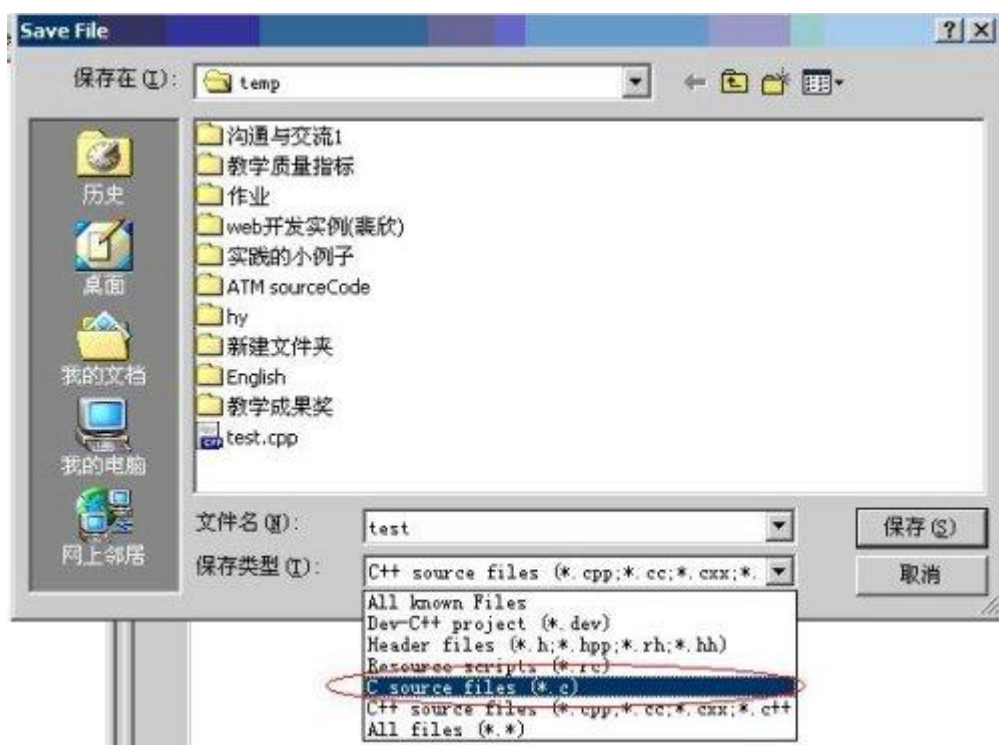


图 5

4. 在程序编辑区域编辑程序，如图 6 所示。在输入程序的过程中记得要随时对程序进行保存（使用菜单“File”→”Save”，或者用组合键 Ctrl+s），此时会将程序重新保存到之前指定的目录下，如 F:\temp。如果想将程序保存到其他的路径下，可以选择“File”→”Save As...”，如图 7 所示，你可以重新指定程序的名称和保存路径。

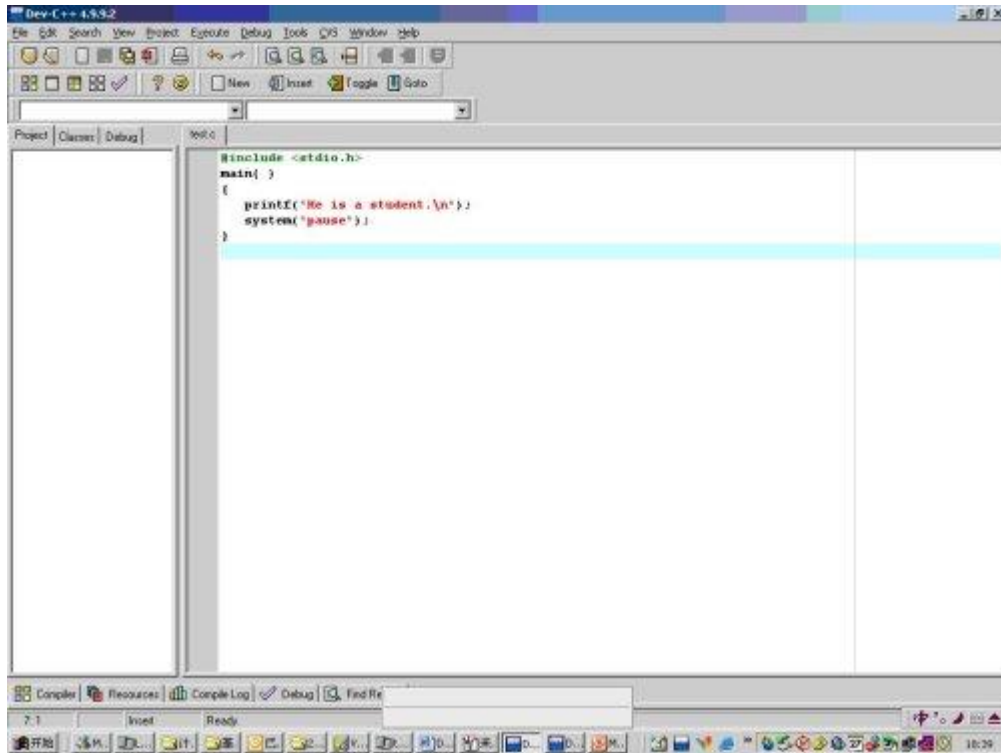


图 6

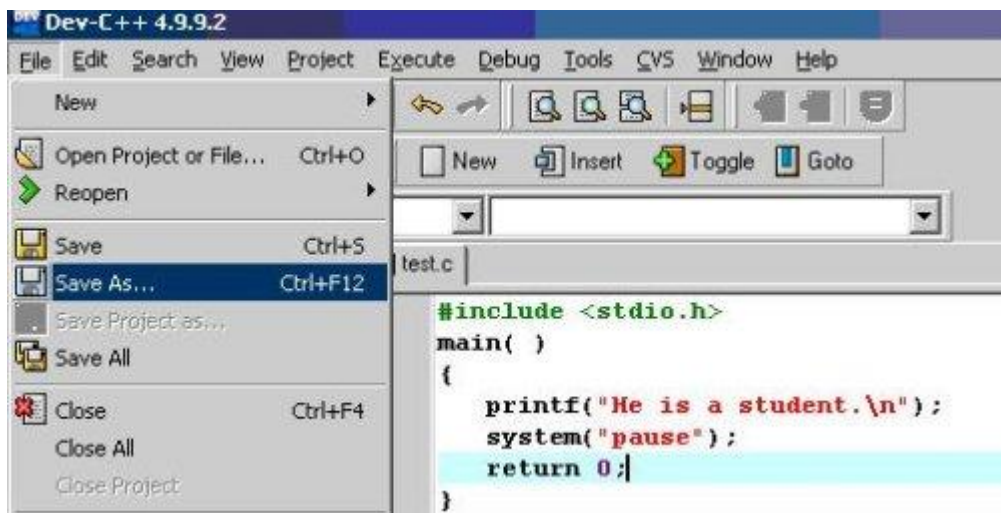


图 7

注意：

1. 必须在**英文**输入环境下编辑程序（如果你当前能在程序编辑区输入中文，说明你是在中文输入环境下。为了输入程序，你必须切换到英文输入环境下）。
2. 在 Dev-C++环境下，为了查看程序运行结果，需要在 main 函数的 return 语句前加上：`system("PAUSE")`或 `system("pause");` 这样程序运行到该语句时，结果显示屏幕将会停留，让大家有时间看程序的输出结果。否则结果显示屏幕将会一闪而过。

三. 预处理、编译、链接程序

从主菜单选“运行”—>“编译”（也可选编译当前文件）或快捷键“Ctrl+F9”，可以一次性完成程序的预处理、编译和链接过程。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“Compile Log”标签页中显示错误信息，如下图 8 所示，并且将源程序相应的错误行标成红色底色，如图 9 所示（由于删除了 `printf` 语句后面的分号，编译时报错，提示 `system` 语句前面的语句有语法错误(syntax error)）。



图 8

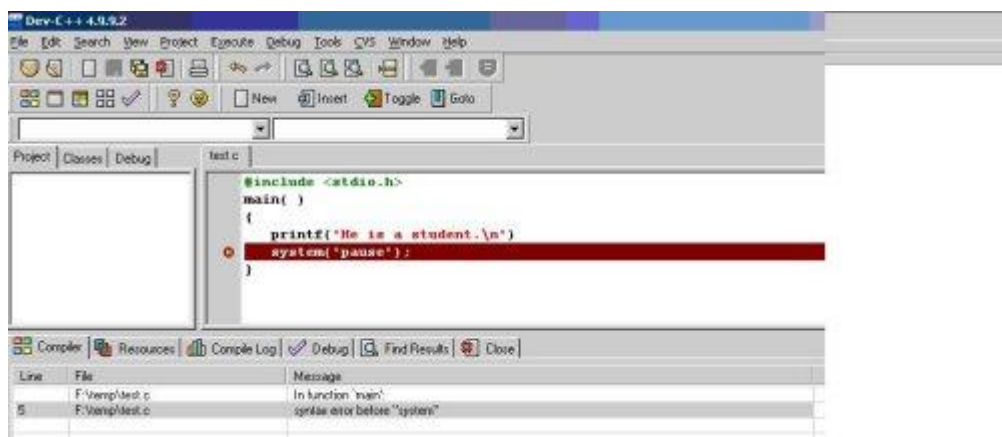


图 9

“Compile Log” 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，每一位同学都要学会看这些错误信息，并且每一次你碰到错误并且最终解决了错误时，要记录错误信息以及相应的解决方法。这样以后看到类似的错误提示信息，能熟练反应出是源程序哪里有问题，从而提高程序调试效率。

排除了程序中存在的词法、语法等错误后，编译成功。此时在源文件所在目录下将会出现一个同名的.exe 可执行文件（如 test.exe）。双击这个文件，即可运行程序。

四. 运行程序

对程序进行预处理、编译、链接后，可以有两种方法运行程序。1. 双击生成的 .exe 文件; 2. 直接在 Dev-C++ 环境下从主菜单选 “Excute” —> “Run” 或 快捷键 “Ctrl+F10” 运行程序。如图 10 所示。

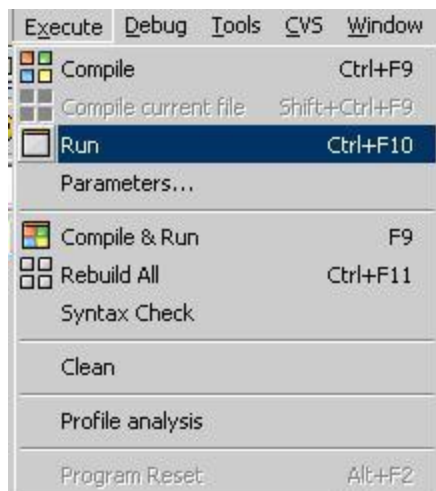


图 10

五. 调试程序

通过预处理、编译和链接的程序仅仅是该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（譬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，**此时需要借助于程序调试（Debug）手段**。这是一种有效的排错手段，每一位同学都需要掌握。

1. 设置程序断点

调试的基本思想是让程序运行到你认为可能有错误的代码前，然后停下来，在人的控制下逐条语句的运行，通过在运行过程中查看相关变量的值，来判断错误产生原因。如果想让程序运行到某一行前能暂停下来，就需要将该行设成断点。具体方法是在代码所在行行首单击，该行将被加亮。默认的加亮颜色是红色。如图 9 所示，将 `system(“pause”)` 语句设成断点，则程序运行完 `printf` 语句后，将会暂停。需要说明的是，你可以在程序中根据需要设置多个断点。

如果想取消不让某行代码成为断点，则在代码行首再此点击即可。

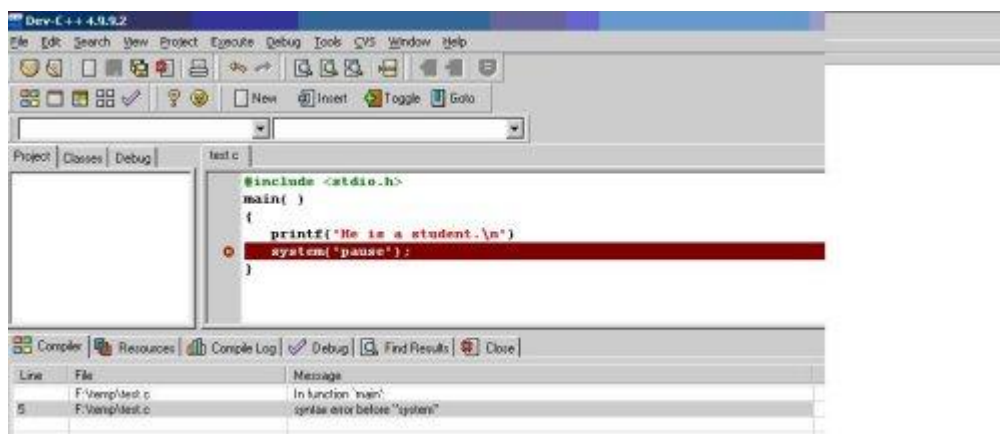


图 9

2. 运行程序

设置断点后，此时程序运行进入 debug 状态。要想运行程序，就不能使用主菜单“Execute”→“Run”，而是需要用主菜单“Debug”→“Debug”（或者按快捷键 F8），如图 10 所示。

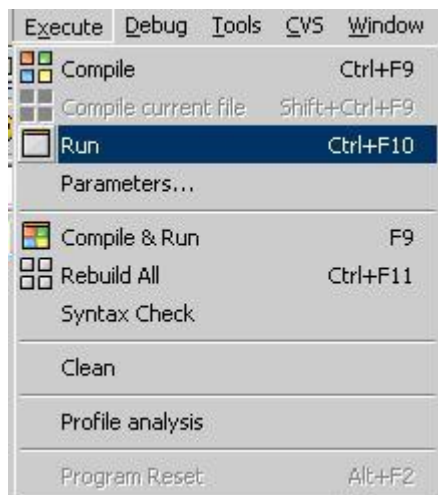


图 10

程序将运行到第一个断点处，此时断点处加亮色由红色变成蓝色，表示接下来将运行蓝色底色的代码，如图 11 所示。

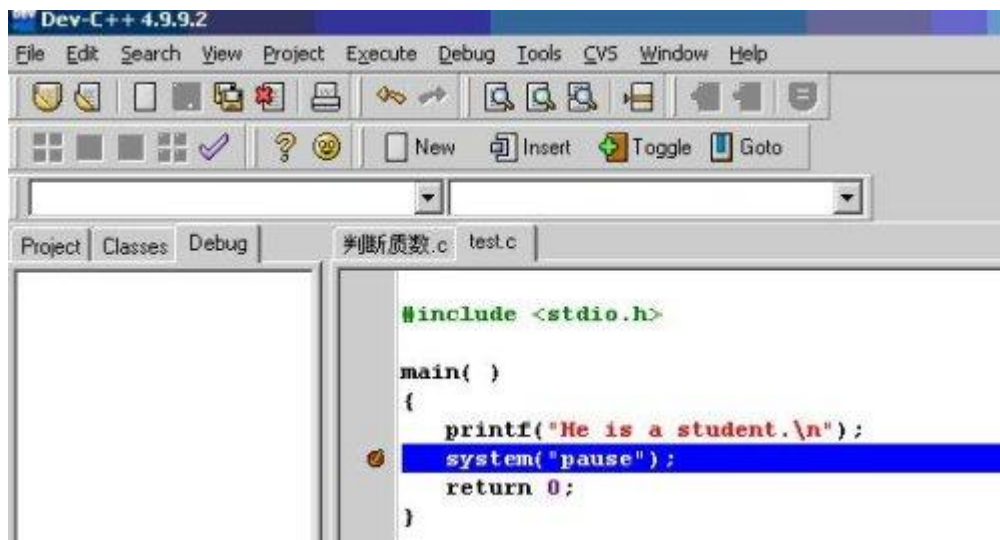


图 11

注意：有时你会发现即使设置了断点，点击了主菜单“Debug”→“Debug”，程序还是不在断点处停留。解决方法：取消断点，重新编译程序，然后再设置断点，点击主菜单“Debug”→“Debug”即可。

3. 单步执行程序

要想运行蓝色底色的代码，可以使用图 10 所示的“Next Step (F7)”、“Step Into (Shift F7)”、“Continue (Ctrl F7)”，“Run to Cursor (Shift F4)”等（具体含义见下面的介绍）。在学习函数之前，一般用的是 Next Step 和 Continue。学习函数后，还会用到 Step Into。

Debug 菜单如图 12 所示。和单步运行相关的菜单项说明如下：

Next Step (F7)：运行下一行代码；如果下一行是对函数的调用，不进入函数体

Step Into (Shift F7)：运行下一行代码，如果下一行是对函数的调用，则进入函数体；

Continue (Ctrl F7)：运行到下一个断点处；

Run to Cursor (Shift F4)：运行到光标所在出；

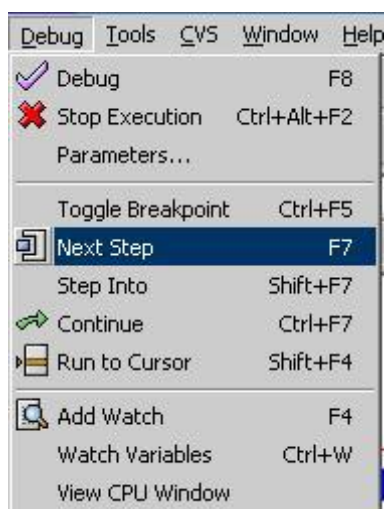


图 12

4. 设置 watch 窗口

在调试程序时，可能要看程序运行过程中变量的值，以检测程序对变量的处理是否正确，可以在调试时 通过调试菜单下的添加变量（Add Watch）窗口来增加变量 watch，新增的变量将会显示在最左边 Explore 的 Debug 页中，如下图 13 所示。如果左边 Explore 中的当前页不是 Debug 页，则可以点击 Debug 标签使之成为当前页。

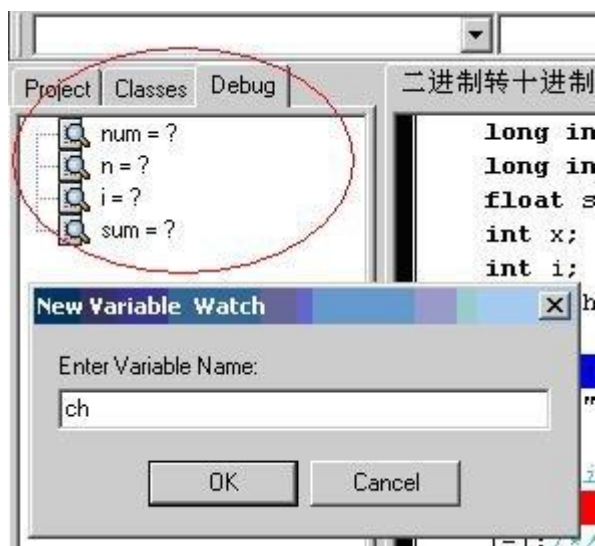


图 13

六. 打开一个已经存在的程序

点击主菜单的” File” ->” Open Project or File” ，如图 14 所示，在弹出的对话框中指定文件所在的路径，选择要打开的文件即可。

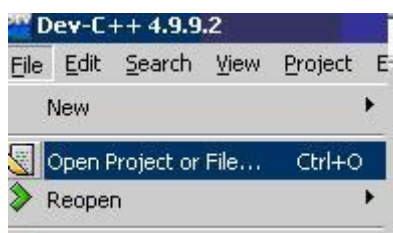


图 14

七. 提高程序书写风格的一些操作

1. 整段缩进

运用适当的缩进，可以提高代码的可读性。选中要缩进的代码段，点击主菜单下的” Edit ” —> “Indent”，如图 15 所示，即可将整段代码右移 N 个字符。

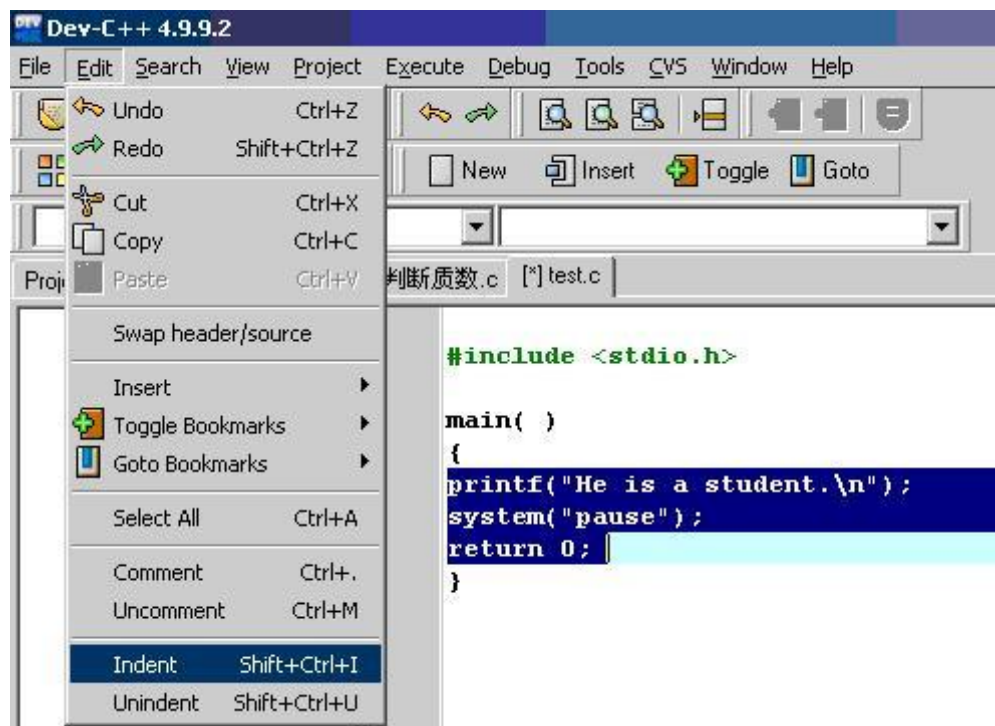


图 15

每一次缩进要移动的字符数可以自己定制，方法如下：点击主菜单下的” Tools ” —> “Edit Options”，在弹出的对话框中的 General 标签页中进行设置，将 Tab Size 设置成你希望的数字，建议设成 3，如图 16 所示。该对话框提供了定制 Dev-C++的界面编辑风格的功能，有兴趣的同学可以自己研究一下。

也可以点击” Edit ” —> “Unindent” 使整段代码往左移。

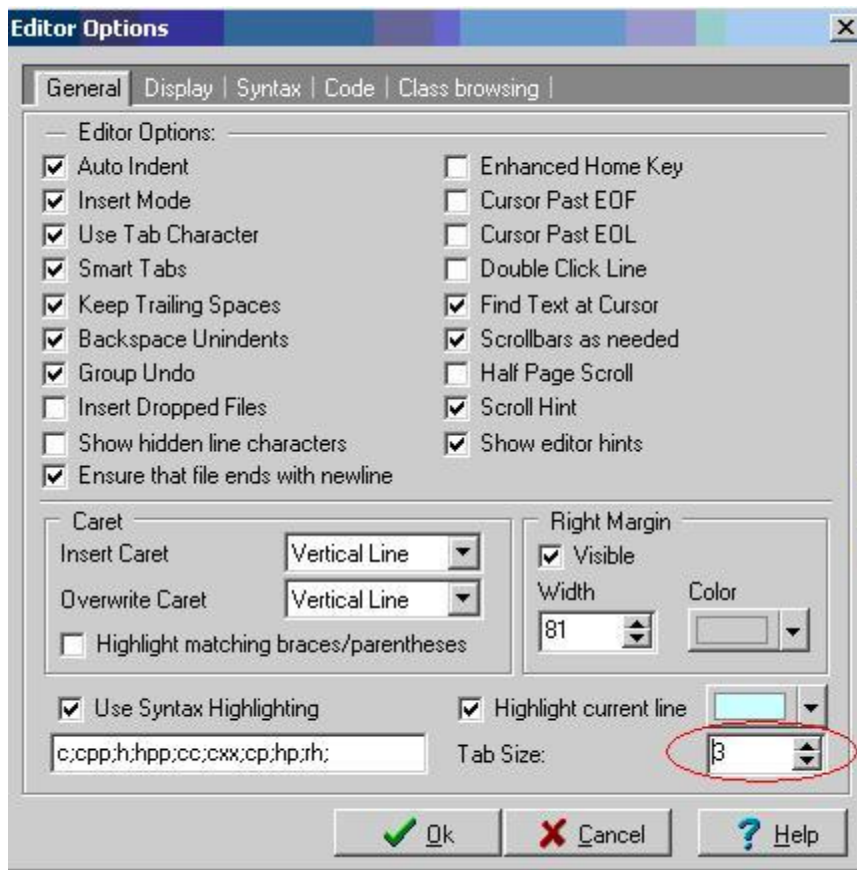


图 16

2. 插入程序说明

在源程序里简要说明程序的功能，是一个良好的习惯，可以点击” Edit ” -> “Insert” -> “Comment Header”，如图 16，从而在程序编辑区光标处插入一段注释，如图 17 所示。

插入后的效果如图 18 所示。

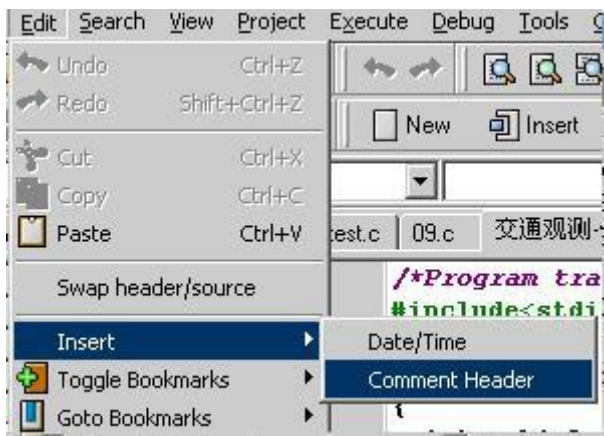


图 17

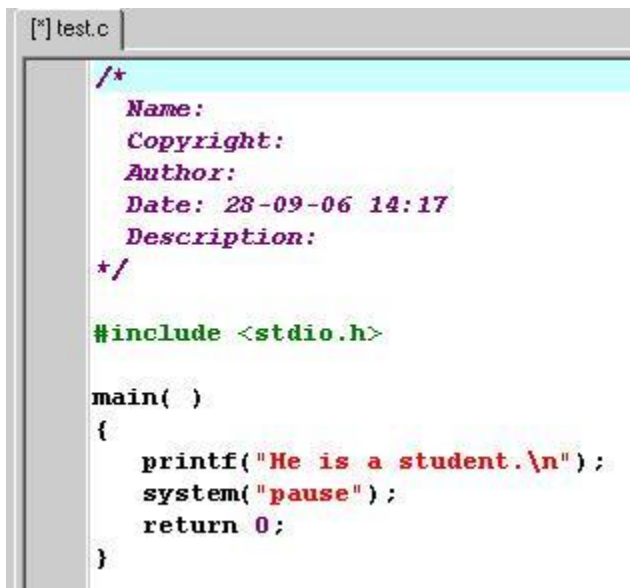


图 18

