Windows 程序设计入门

潘凌云 编著

人民邮电出版社

计算机技术丛书

Windows 程序设计入门

潘凌云　编著

人民邮电出版社

登记证号(京)143 号

内 容 提 要

本书是 Windows 程序设计的初级教程。全书共分 10 章，由浅入深地给出了 Windows 程序设计最基本、最实用的内容。在讲解时，书中给出一些调试过的程序。本书适合读者自学，也可作为大学有关专业及培训班的教材。

计算机技术丛书

Windows 程序设计入门

潘凌云 编著

*

人民邮电出版社出版发行

北京朝阳门内南竹杆胡同 111 号

北京顺义振华印刷厂印刷

新华书店总店科技发行所经销

*

开本:787×1092 1/16 1995 年 3 月 第一版

印张:12 1995 年 3 月 北京第 1 次印刷

字数:290 千字 印数:1—7000 册

ISBN7—115—05537—8/TP・154

定价:16.00 元

丛书前言

世界上发达国家普遍重视发展以计算机和通信为核心的信息技术、信息产业和信息技术的应用，一些经济发达国家信息产业发展迅速。

当前，我国处于国民经济高速发展时期。与此相伴随，必将有信息技术、信息产业和信息技术应用的高速发展。各行各业将面临信息技术应用研究与发展的大课题以及信息化技术改造的大任务、大工程。

为了适应信息技术应用大众化的趋势，提高应用水平，我们组织编写、出版了这套“计算机技术丛书”。这套丛书以实用化、系列化、大众化为特点，介绍实用计算机技术。

这套丛书采取开放式选题框架，即选题面向我国不断发展着的计算机技术应用的实际需要和国际上的实用新技术，选题不断增添又保持前后有序。

这套丛书中有的著作还拟配合出版软件版本，用软盘形式向读者提供著作中介绍的软件，以使读者方便地使用软件。

我们希望广大读者为这套丛书的出版多提意见和建议。

前　言

目前，Windows 系统在微机上得到了广泛的应用，一个新的 Windows 微机操作系统——Windows NT 也已经问世。从目前微型计算机的发展趋势来看，Windows 系统在微机上取代 DOS 已是必然的趋势。可以说，微机发展史上的一个新的时代—— Windows 软件时代已经到来。

越来越多的人希望了解 Windows 系统的有关内容，而一般的关于 Windows 程序设计的书大多为手册形式，其内容多而全，不适于初学者学习。本书是一本入门性质的教程，结合作者的实际编程经验，具有较强的实用性。全书简明易懂，详细介绍了在 Windows 3.1 系统下使用 C 语言进行程序开发的基本知识和技术。全书分成十章，由浅入深逐步给出了程序开发中最基本、最实用的内容。本书包含了许多实际例子的源程序清单，以帮助读者进一步理解这个系统下的程序开发方法。所有例子程序都经过上机调试，是可直接运行的。

本书的读者对象是希望了解 Windows 系统的业余计算机爱好者和计算机程序员，也可以作为大学有关专业学生的自学书或参考教材。

学完本书之后可以使读者初步掌握 Windows 的程序开发技术，并能编写一些简单的 Windows 应用程序。

为顺利理解这本书，我们希望读者具有初步的微机操作经验和 C 语言编程知识，以及有关 Windows 系统的初步使用经验。

最后，我要感谢孙达传教授，他审校了本书，并提出了许多宝贵意见。

1994 年 10 月

目　录

第一章 Windows 程序开发基础知识

1.1 什么是 Windows

Windows 系统是 Microsoft 公司为微机操作系统开发的图形用户介面，它把三个重要的功能——图形用户介面、多任务和硬件的独立性集成到操作系统环境中。该环境为用户程序提供了基于鼠标器控制的统一的窗口和菜单介面。与传统的 DOS 操作系统环境相比，Windows 操作环境对用户来说具有很大的优越性，更易于学习和使用，使软件开发者能又快又容易地开发出用户友善的应用软件。

Windows 系统 3.1 版是在它的前身 Windows 3.0 版的基础上作了很多改进后于 1992 年推出的，这个新版本被认为是图形用户介面的标准，有越来越多的基于这个标准的应用软件正在出现并得到应用。目前，Windows 3.1 系统在微机上得到了广泛的使用，一个新的 Windows 微机操作系统——Windows NT 也已经问世。并且，大有取代 DOS 操作系统的趋势。可以说，微机发展史上的一个新的时代——Windows 软件时代已经来到。

计算机的处理能力在日新月异地发展，但是微机上原有的 DOS 操作系统却无法跟上这个发展速度。例如，对于多任务处理能力以及虚拟内存管理能力等等。原有的 DOS 操作系统只能通过一些非直接手段去实现它们，应用程序的运行方式仍然是基于 DOS 方式或称为实模式方式上的想法。这样的状态大大限制了应用软件的发展。为克服这一系列的缺点，引入了 Windows 系统。并且，这个因素也决定了 Windows 系统代替 DOS 系统的必然性。

1.2 Windows 应用软件与 DOS 应用软件的比较

Windows 3.1 系统具有很多 DOS 系统下所没有的特征，所以在一些方面 Windows 应用软件更加复杂。这些特征包括：

(1) 用户程序都提供具有窗口和菜单形式的图形介面。

(2) 允许多任务运行。

(3) 使用消息队列来实现事件驱动的程序运行方式。

(4) 实用程序间的动态数据交换。

(5) 与设备无关的图形功能和打印功能。

接下来，让我们看看开发一个窗口软件与开发一个标准 DOS 软件的区别。这里我们以 C 语言为例。在 DOS 环境下，用户程序使用标准 C 语言运行库中的函数实现程序的输入/输出、直接的存储管理等，但在 Windows 环境下这些手段就不再有效了。因为在 Windows 系统里应用程序要以多任务方式共享各种资源，而不能独占这些资源。另外，显示器、键盘和鼠标器也是为多个用户程序所共享的。更重要的是，用户程序不能随便地对内存单元进行直接处理，这是

由于内存区域是由多个用户所共享的，程序和数据可以在内存中按需要进行移动，从而实现内存区的重新分配。

在开发一个标准的 DOS 软件时，一般有以下几个步骤：先是编写源代码，然后调用编译器进行编译形成目标文件，最后，用连接器 LINK. EXE 把目标文件与库文件连接成一个可执行文件。图 1.1 说明了开发一个 DOS 应用程序的过程。

而开发一个窗口应用软件还需要增加一些额外的编程和准备工作。这些工作包括：

(1) 源程序中包含一个特殊的头文件 windows. h，它含有各种有关窗口的数据类型，常数，函数原型等信息。

(2) 与窗口有关的库文件。

(3) 一个模块定义文件(. def)用于连接程序目标块和库函数。

(4) 一个资源文件(. rc)描述了应用软件所需的各种资源，它含有图、字符串、菜单等。一个资源编译器 RC. EXE 会对它进行编译并与. EXE 文件进行连接。

(5) 用 LINK. EXE 连接好的可执行文件还不能在 Windows 下正常运行，必须把它与资源进行连接后才能生成一个 Windows 应用软件。

图 1.2 说明了开发一个 Windows 应用程序的过程。

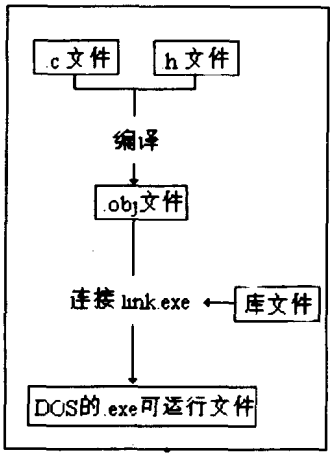

图 1.1 DOS 下 C 语言开发程序过程

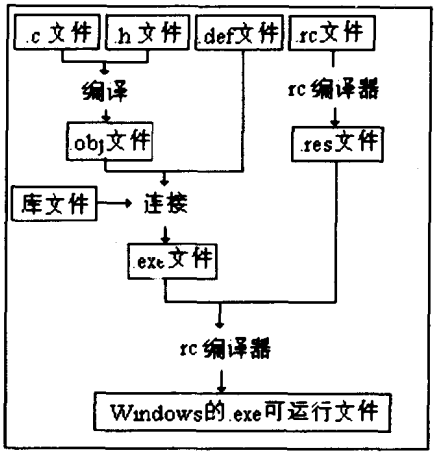

图 1.2 Windows 下 C 语言开发程序过程

1.3 Windows 应用软件的组成部分

由图 1.2 可以看到，一个 Windows 应用程序除了 C 语言源程序 (. c) 文件以外，还要有一些附加文件。主要有模块定义文件 (. def) 和资源文件 (. rc)。除此之外还要有一些用户头文件 (. h) 和窗口库文件 (. lib 和 . dll)。

1.3.1 库文件

所有与窗口有关的系统函数和它们的定义都存放在动态连接库文件中，它们的文件名以

dll 为其扩展名(或者以 exe 为其扩展名)。与标准 DOS 下 C 语言的运行时间库不同的是,Windows 使用了动态连接库技术(DLL)。它的实际含义是,应用程序用到的被调用函数只有当软件装入内存运行时才与应用程序连接起来。并且,这样的动态连接库函数在内存中只有一个备份,每个要使用同一函数的程序都与这一函数的备份进行连接。这个技术的好处是少占内存,并且使库中函数的更新与应用软件彻底分开。

Windows 系统提供以下三个动态连接库:

(1) KERNEL.EXE 用于多任务、内存管理和资源管理。

(2) USER.EXE 处理窗口管理和输入。

(3) GDI.EXE 提供图形介面和输出。

所有这些动态连接库中的函数都可以提供给用户调用。为了顺利地实现程序的编译和连接,有关这些函数的调用原型信息存放在头文件 windows.h 中,其连接信息存放在库文件 .lib 中。这些库文件随C 编译器的不同而有所差别。在 Borland C 编译器下,它包含了与窗口有关的库文件:bwcc.lib, cw?.lib, c0w?.lib, mathw?.lib 等(其中的问号 "?" 可以是以下四个字符中的一个:s,m,c,l。它们分别代表 Windows 系统的四个存储模式)。

1.3.2 Windows 的函数

所有与 Windows 有关的应用程序接口(API)函数都包括在上面的三个动态连接库中,依次分成三类:系统服务接口函数、窗口管理接口函数和图形设备接口函数。后面我们将会逐渐遇到很多这三类函数。在附录 A 中,列举了其中常用的一些应用程序接口函数名,对于每个函数具体使用格式和参数类型,可以在编程序时使用在线帮助选择项 help 来查询。

对于每个 Windows 应用程序的源程序,必须含有以下两个基本函数:

(1) 主函数 WinMain()。它代替了通常 C 程序中的 main()函数的地位和功能。

(2) 主窗口过程函数。每个程序要有一个用户窗口,而每个用户窗口则必须要有一个这样的窗口过程函数,用来处理所有有关这个用户窗口的操作。例如,在例子程序 example1.c 中的函数 exampleWndProc()就是这个用户窗口的窗口过程函数。这个函数的名字可以任取,它完成了用户窗口的基本操作功能,例如窗口用户区信息的显示、各种事件对应消息的处理等。

1.3.3 自定义数据类型

与标准 DOS 下的 C 语言不同,Windows 3.1 中有许多自定义数据类型,它们都在头文件 windows.h 中给出了定义。主要有如下一些基本数据类型:

类 型	意 义
BOOL	布尔类型量,占据 16bit
BYTE	字节类型,无符号 8bit 值
WORD	字类型,无符号的 16bit 值
DWORD	双字类型,无符号的 32bit 值
UINT	无符号整数类型 unsigned
LPSTR	指向字符串的长指针
HANDLE	一个通用句柄类型,为 16bit 整数值
HDC	指向设备描述表的句柄类型,16bit 整数值
HWND	指向一个窗口的句柄类型,16bit 整数值

上面提到的句柄是一个新的概念，在 1.4 节中将给出说明。

1.3.4 标记规则

在本书中我们使用匈牙利标记法来定义变量名或其它的标识名，其主要思想是利用变量名开头的一个或两个字母来表明这个变量的数据类型，变量名的剩余部分将描述该变量的功能。这个方法已被广范应用于程序开发工作中，使程序易于阅读。例如，用 hWnd 表示一个窗口的句柄，其中首字母 h 代表句柄 (handle)，Wnd 代表关于窗口 (window)的变量。另外，对在一个标识名中出现的每个单词的首字母采用大写字母。故当用户知道这一方法后，就很容易从变量名推断出它的含义来。这就是匈牙利标记法。下面列举了本书中出现的一些变量首字母和它们的意义：

首字母	意义
b	BOOL 布尔变量
by	BYTE 型变量
c	char 字符型
i 或 n	int 型变量
l	long 型变量
sz	以 NULL 结束的字符串变量
w	WORD 型变量
dw	DWORD 型变量
fn	函数型变量 function
lp	长指针型变量 32bit
h	16bit 句柄变量
pt	坐标变量 POINT
r	矩形变量 RECT

1.4 一些新的概念

1.4.1 窗口

窗口是运行程序与外界交换信息的介面，它是屏幕上的一块矩形区域。每一个应用程序一般都至少拥有一个窗口，作为自己的信息交换介面，有时可以拥有多个窗口。另外，Windows 下的各种软件虽然具有不同的功能，但都有十分类似的窗口和菜单形式，以方便用户的使用。图 1.3 是一个典型的窗口形式。

下面将对这些概念逐条加以说明：

系统菜单——系统菜单由 Windows 系统预先定义，并始终出现在窗口的左上角，当用户用鼠标对准这个区域并揿下鼠标器的最左键时会出现一个下拉菜单，其中有多个功能选择项，它们包括：移动窗口、改变窗口大小、关闭本窗口、切换到其它窗口等等。

标题栏——能显示一条说明信息，用来表示本窗口功能特点。当这个区域的背景色为深颜

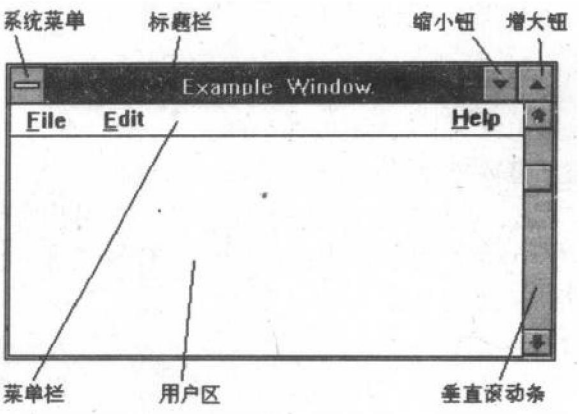

图 1.3 一个窗口的形式

色时，表示这个窗口是当前活动窗口。

菜单栏——显示用户菜单的第一级菜单选择项，当用户用鼠标选定某一个选择项时可以出现这个选择项的二级菜单，从而用鼠标可以选择菜单中的任一选择项。当一个选择项名字中某一字母有一下划线时，表明可以用 Alt＋<字母键>来选定这个选择项。

用户区——对用户来说这是一个十分重要的区域，是主要的信息输入/输出区域。

缩小钮——当用户用鼠标对准这个区域，并撤下鼠标器的最左键时，窗口缩小尺寸或者直接缩小成一个图标。

增大钮——当用户用鼠标对准这个区域并撤下鼠标器的最左键时，窗口会增大到最大的可能尺寸。

垂直滚动条——它含有一个标有向上箭头的上滚框、一个标有向下箭头的下滚框，以及一个表示位置的位置小方块。每当在上滚框或下滚框上撤一下鼠标器最左键时，用户区的显示内容会上滚或下滚一行，每当在位置小方块与上滚框或下滚框之间撤一下鼠标器最左键时，用户区的显示内容上滚或下滚一页。若用鼠标器直接选择位置小方块，并拖动它到任一新的位置时，则可直接选择文件的某一个显示内容。

1.4.2 窗口类(windows class)

窗口有许多不同的表现特征，如窗口的式样、颜色、光标的式样、图标的式样等等。具有相似表现和特征的窗口属于同一个窗口类。并且，每一个窗口一定属于某一类，也仅仅属于一个类。用户可以定义自己的窗口类。另外，Windows 系统还提供了几个预定义窗口类，这在 6.7 节中将会给出祥细说明。

1.4.3 句柄(handle)

句柄是 Windows 系统提出的一个重要的概念。它是一个数据类型，用 16bit 表示的整数。Windows 系统用一个无重复整数来标识应用程序中的一个对象，如窗口、菜单、内存区域、文件等等。Windows 把与对象有关的信息存放在一张内部表格中，而用句柄来指明每个对象在表格中的入口地址。应用程序只能通过句柄来访问所指定的实际对象，不能直接访问，这个性质是由 Windows 的多任务环境所决定的。

1.4.4 图标(icon)

这是一个图形小标记，用来表示文件、运行的程序、特殊警告信息等。用图标作为这些对象的标记，其优点是更能引起用户的注意，容易理解其含义。Windows 系统提供了几个预定义的图标，如问号、惊叹号、小方块等。另外，还允许用户设计自己定义的图标。

1.4.5 光标(cursor)

与 DOS 环境下的光标不同，这里的光标是一个图形符号，它对应于鼠标的位置。常见的光标有：

箭头光标——此时，用户可以移动鼠标器来选择对象。

沙漏形光标——表示系统正忙于某项任务，鼠标被锁住。

十字准线——表示可以移动当前窗口的位置。

水平箭头/垂直箭头——表示可以改变窗口的大小尺寸。

另外还有几种预定义光标。同时，Windows 系统也允许用户设计自己定义的光标。

1.4.6 字符插入标(caret)

插入标是一个可闪烁的短线条，表示在这个位置可以插入新的字符。它的作用类似于 DOS 环境下的光标。

1.4.7 剪接板(clipbord)

剪接板实际上是一块内存缓冲区域，用来存放一段字符串、一块图象或任一格式数据，它的用途是为不同程序或窗口之间实现数据交换。一些应用程序，如编辑软件中的删除、复制和插入等功能都是通过剪接板实现的。

1.4.8 实例(instance)

一个应用程序能在 Windows 多任务环境中同时运行几个副本。例如，可以同时在屏幕上显示几个相同的时钟，即多次运行 clock.exe。一个程序的每次运行被称为是这个程序的一个

实例。为区别相同程序的几个实例，每个实例在启动时都由系统分配给一个互不相同的实例句柄（hInstance），这是一个用 16 bit 表示的整数。与其它的多任务系统不同，在 Windows 系统中，一个程序的多个实例都使用相同的程序代码，其区别之处是每个实例加载不同的数据段。这样，每个程序代码在内存中只保留一个副本。此方法的好处是节约内存的占用量。

1.4.9 注视窗口

Windows 系统始终知道哪一个窗口当前拥有键盘和鼠标，Windows 系统必须把鼠标器或键盘发出的事件对应的消息传给当前拥有鼠标的应用程序。通常，鼠标器的光标所在的窗口将拥有鼠标，这个窗口将获得鼠标和键盘的所有有关消息。这个窗口被称为注视窗口。我们能从窗口标题栏的颜色来识别一个活动窗。通常，一个活动窗就是注视窗口。用户可以用组合键 Alt＋Esc，或 Alt＋Tab，或者用鼠标器来实现注视窗口的转移。

1.5 Windows 环境和设置

在进行 Windows 3.1 软件开发工作之前，我们必须选择一个合适的计算机系统的硬件和软件环境，否则将不能发挥 Windows3.1 的强大功能。我们推荐用户选择以下的系统配置：

（1）具有 Intel 80386/80486 微处理器的个人电脑，当选用 80386SX 芯片时，Windows 的运行效果不太好，建议使用于 80386DX 以上的芯片。

（2）至少 2MB 以上的内存，最好采用 4MB 以上内存，这样有利于多任务运行窗口间切换的速度。

（3）DOS 5.0 以上版本的操作系统软件。

（4）1.44 Mb 3″ 软驱一个，最好再有一个 1.2Mb 5″ 软驱，这样有利于各种软件的安装。

（5）120Mb 以上容量的硬盘一个。由于 Windows 下的软件，其尺寸都比较大，故要选大容量硬盘为宜。并且注意当 Windows 运行时，要留有 10Mb 左右的可用硬盘空间，以便系统的正常工作。

（6）VGA 或 SVGA 彩色或单色显示器一个，若再增加一块图形加速卡，则对窗口的切换速度更有帮助。

（7）机械式或光电式鼠标器一个，其型号必须是 Windows 系统中已带有驱动程序的那些型号之一，这些型号可由系统配置程序 setup.exe 来查得。

（8）EPSON 或 HP 打印机一台，或其它类型的 Windows 带有驱动程序的打印机。

在 Windows 3.1 运行之前，还必须检查一下你的 DOS 系统配置文件 config.sys，它一般应包含以下内容：

```
files=30
buffers=30
device=himem.sys
device=smartdrv.sys 1024 512
device=emm386.exe
```

其中的第三行是必不可少的，它保证了 Windows 系统能使用 1Mb 以上的内存区域。若你的计

算机内存大于 4Mb，则在你的 DOS 系统配置文件 config. sys 中可以包含上面的第四行来建立 CPU 的高速缓存区功能，其大小由后续的二个参数给定。注意，在 DOS 6.0 以上版本中这个文件的名字改为 smartdrv. exe。上面的第五行是建立扩充内存功能，是否要加入这一项，取决于实际应用程序的需要。

当证实你的配置文件正确后，可以启动 Windows，否则先修改配置文件 config. sys，然后重新启动计算机一次。完成以上工作后，可以用以下命令启动 Windows 3.1 系统：

```
>cd  windows
>win  (或>win/3)
```

最后要注意的是，Windows 系统的初始环境参数文件有 WIN. INI，SYSTEM. INI，PROGRAM. INI 等等，它们给出了当前 Windows 系统的环境参数。其中只有 WIN. INI 可以由用户用正文编辑软件改动之，其它的初始参数文件都不应当直接改动之。

1.6 程序开发工具

程序的开发还需要有一个 C 语言编译器。用户可以使用 Borland C++ 3.1 编译器，或者使用 MS C++ 7.0 编译器来实现程序的开发。

使用 Borland C++ 3.1 编译器作为程序开发环境时，它具有以下窗口软件开发工具：

(1) 集成开发环境 bcw. exe。

(2) 资源编译器 rc. exe。

(3) 窗口有关的头文件 windows. h 和库文件 bwcc. lib，cw?. lib，c0w?. lib，mathw?. lib 等等。

(4) 资源编辑工具 workshop. exe。

(5) 查错工具 winspctr. exe。

(6) 窗口和消息观察器 winsight. exe。

使用 MS C++ 7.0 编译器时，Microsoft 公司提供了窗口软件开发工具(SDK)3.1 版，它包括了许多有用的编程工具和辅助工具，其中有：

(1) 连接器 link. exe。

(2) 资源编译器 rc. exe。

(3) 可见代码调试程序 cvw. exe。

(4) 窗口有关的头文件 windows. h 和库文件? libcew. lib，libw. lib 等等。

(5) 图象编辑器 imagedit. exe。

(6) 字体编辑器 fontedit. exe。

(7) 对话框编辑器 dlgedit. exe。

本书的例子程序都是使用 Borland C++ 3.1 编译器进行调试和实现的，但这些程序完全适用于 MS C 7.0 编译器。

1.7 Borland C 版本 3.1 编译器的使用

本书的所有例子程序都是用 Borland C 3.1 版进行编辑和编译的，建议用户也使用这个 C 语言开发工具。它提供给用户一个强有力的集成开发环境界面 bcw.exe，其大部分操作习惯都与 Turbo C 集成开发环境类似。用户可以在进入 Windows 系统之后，在文件管理器窗口下运行\borlandc\bin 目录下的文件 bcw.exe，即可进入 Borland C 3.1 版的集成开发环境。

进入 bcw.exe 集成开发环境后，可以进行源程序编辑、编译、连接和运行工作。在做这些工作之前用户还必须先完成两件事情：

(1) 设置 Borland C 集成开发环境的参数。在这个集成开发环境窗口的菜单栏选择项 Options 下，设置一些必要的参数，其中包括：库文件路径(例如 C:\borlandc\lib)、包含文件路径(例如 C:\borlandc\include)，以及源程序和输出文件的路径。接着，要设置连接后的运行程序的存储模式，选择以下四个模式之一：large，medium，compact 和 small。还要设置运行目标的类型，可以是 Windows 的可运行文件.exe 或者是动态连接库文件.dll 类型。一般情况下以 exe 类型为缺省类型。

(2) 建立一个与应用程序同名的.prj 文件，这是源程序的编译和连接所必须的。在集成开发环境窗口的菜单栏选择项 project 下，可以建立一个新的 project 文件(.prj)，然后逐项加入与这个应用程序有关的各个源程序文件名(.c)、模块定义文件名(.def)、资源文件名(.rc)，若有其它的目标文件(.obj)需连接，则也需加入进去。与 Turbo C 中的 project 文件不同，这里的.prj 文件不是正文文件，只能在集成开发环境的 project 选择项下生成和修改。

当完成以上的工作后，用户可以在这个集成环境的编辑窗口中编辑 C 语言源程序。这是一个多窗口编辑环境，可以同时打开和编辑多个源程序，以及方便地进行文件之间的数据交换。程序的编译、连接和运行工作，可以选定菜单栏的运行 Run 选择项来完成。若仅仅做编译工作则可以选定菜单栏的编译 Compile 选择项来完成。总之，这个集成开发环境是目前 PC 机上最完美的开发环境之一，用户一经使用这一工具就会被它的功能所吸引，成为 BorlandC 的忠实用户之一。