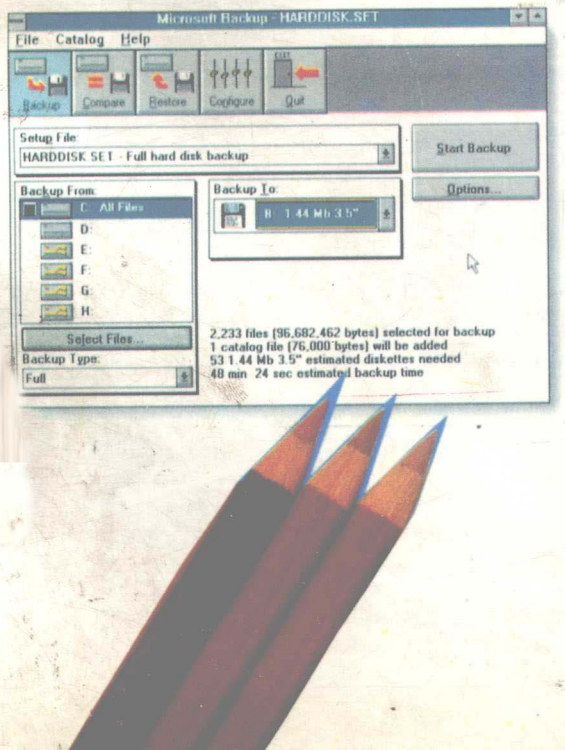
Windows技术丛书之二

Windows 编程基础

李 力 编著

海洋出版社

2



Windows 技术丛书之二

Windows 编程基础

李 力 编著
文 枫 审校

海洋出版社
1993 年·北京

内容提要

本书描述了由 Microsoft Windows 操作系统所支持的不同的接口函数和扩充库，书中还有关于应用的专用窗口特性说明。

附录列出了有关窗口函数的模块软件及库名清单。

欲购本书的用户可直接与北京 8721 信箱联系，邮编：100081，电话：2562329。

（京）新登字 087 号

责任编辑　闫世尊

Windows 软件技术丛书之二

Windows 编程基础

李　力　编著

文　枫　审稿

海洋出版社出版（北京复兴门外大街 1 号）

海洋出版社发行　兰空印刷厂印刷

开本 787×1092 1/16　印张：19.875　字数：459 千字

1993 年 12 月第一版　1993 年 12 月第一次印刷

印数 1—5000 册

*

ISBN 7—5027—3821—5/TP. 234　　定价：193.00 元/套（6 册）

Windows 软件技术丛书

1. 高级编程指南	29.00 元/册
2. Windows 编程基础	29.00 元/册
3. Windows 函数速查手册	66.00 元/册
4. MS-DOS 和 Windows 环境下的 C++ 语言编程指导	25.00 元/册
5. MS-DOS 和 Windows 环境下的 C 语言编程指南	19.00 元/册
6. MS-DOS 和 Windows 环境下的 C/C++ 编程技术	25.00 元/册

前　言

本书首先描述了窗口管理、图形和系统服务，列出了与窗口管理、图形输出及系统服务，窗口管理器函数进程消息，建立、移动、或切换一个窗口，建立系统输出等方面有关的函数。图形设备接口（GDI）函数完成与设备无关的图形操作，如在不同输出设备上产生线段、正文和图形输出。系统服务函数完成如下操作：访问模块中的代码及数据，分配和管理内存，解释字符串，建立和打开文件。

然后介绍扩充库，提供了许多种支持 Windows 3.1 新的特性的库。这些新的特性包括公共对话框，简化动态数据交换的管理功能（DDE）对象链接与嵌入（OLE），Shell 功能增强如登录数据库、拖拉特性，工具求助功能使得窗口内工具的建立过程变得连贯，数据压缩功能，一种用于人工评估系统资源及应用查错的重点试验工具，文件安装功能如允许应用程序在 80386 及 80486 处理器上使用 32 位的内存寻址能力，浮点仿真，包含于控制板中的屏幕保存器等。

最后是应用说明，描述了应用程序应该采用的 Windows 特性及增强功能。这一部分解释了如何建立控制版应用程序，如何建立和安装文件管理器的扩充部分，如何使用程序管理器中动态数据交换接口，怎样建立与地区及语言无关的应用程序，怎样编写网络应用程序，如何将窗口应用与 MS-DOS 函数集成在一起，如何书写生成 Windows 前置码及后续码的编译器，怎样初始化及启动窗口应用程序，如何改进窗口系统的视觉效果，怎样编写自装载窗口应用程序以及如何与驱动器交互。

附录列出了每一窗口函数的模块及库名清单。

本书内有关符号含义如下：

符号含义：

下面约定在全手册中通用。

符号	意义
黑体字段：	表示一个术语及某一量的符号，如资源定义说明，函数名（MENU、CreateWindow）、MS-DOS 命令、指令行选项（/nod）。用户应用时必须按书中所示键入字符。
斜体字段：	表示一个变量，用户需提供其实际值。如声明SetCursorPos (X，Y)，就要求用户

以实际值替换 X 和 Y。

[ ]：	可选参数
|：	分离那些选项
...：	表示以下的内容可能是重复的。
BEGIN…END	表示一应用示例的省略部分

除此之外，某些正文符号用来帮助读者理解本书内容。

符号	意义
小号大写	表示键名、键顺序以及键的组合一如ALT+SPACEBAR。
大号大写	表示文件名及路径，许多类型及结构名（同时也是黑体），常量。
单个空格	分隔符。

目　录

第一章　窗口管理

本章描述 Microsoft Windows 操作系统中的一些函数,它们处理消息，建立、移动和改变窗口，建立系统输出．这些函数组成了窗口管理程序接口．

1.1　消　　息

消息是应用程序的输入，表示需要应用程序响应的事件．消息是一个包含消息标识符和信息参数的结构．参数内容随消息类型而变．

1.1.1　消息的产生和处理

窗口为每一输入事件生成输入消息，如用户移动鼠标或按下键盘．窗口在系统级消息队中收集输入消息，然后将消息、定时器和显象信息置入应用消息队列中．应用消息队列除定时器和显象消息外，都是先入先出队．定时器和显象消息在应用程序处理完所有其它信息前存于应用信息队列中．窗口将属于专用应用软件的消息置入那一应用消息队列中．应用程序利用 GetMessage 函数读取消息，并通过 DispatchMessage 函数分配它们到相应的窗口程序中．

窗口直接发送消息到窗口程序中，而不是将该消息置入应用信息队列中，这些消息称为非排队消息．一个非排队消息通常只影响窗口．由 SendMessage 函数直接发送消息到窗口程序中．有关窗口程序的详细情况，参见 1.2.13“窗口程序”。

例如 CreateWindow 函数告诉窗口发送 WM__CREATE 消息到一个应用的窗口程序中，然后等待，直到窗口程序处理完消息．窗口直接发送这个消息到窗口程序而不是将其置入应用消息队列中．

尽管窗口生成许多消息，但是一个应用程序能够生成它自己的消息，并且将它们放入自己的消息队或其它的应用消息队列中．

应用程序通常在一个 WinMain 函数循环中使用 GetMessage 函数转移应用消息队列中的信息．这一循环叫做主消息循环．GetMessage 函数搜索应用消息队，如果消息存在，则返回队列中的顶部消息．如果消息队空，GetMessage 函数则等待直到有消息置于队列中．GetMessage 函数在等待时放弃对窗口的控制，允许其它应用程序控制窗口并处理自己的消息．

一旦应用程序 WinMain 函数从应用消息队列中访问到消息，就使用 DispatchMessage 函数分配消息到窗口程序中．这一函数通知窗口调用与消息有关窗口的窗口程序，然后将消息内容作为函数自变量传递给所调用的程序，窗口退还在 WinMain 函数中的控制权给主消息循环．然后，主消息循环访问队列中下一个信息．

除非特殊说明，窗口可以按照任意顺序发送消息．应用程序可以不依赖于以某种接收消息的顺序．每当用户按下一个键窗口就生成一个消息．消息包含识别键盘的虚拟键码，

但不定义该键的字符值。如果要访问字符值，WinMain 函数中的主消息环必须使用 TranslateMessage 函数解释虚拟键消息。该函数将带有相应字符值的消息放到应用消息队列中，这些消息随后被分配到窗口程序中。

1.1.2 解释消息

一般而言，WinMain 函数使用 TranslateMessage 函数解释每一个消息，而不仅仅解释虚拟键消息。尽管 TranslateMessage 函数对其它类型的消息没有影响，但它可以保证正确地解释键盘输入消息。

下面例子表示了典型的主消息循环，在这一循环中，WinMain 函数用于访问应用信息队列中的消息，并分配这些消息到应用窗口程序中。

```
int PASCAL WinMain(hinst, hPrevInst, lpCmdLine, ShowCmd)
HINSTANCE hinst;
HINSTANCE hPrevInst;
LPSTR lpCmdLine;
int ShowCmd;
{
    MSG msg;
     .
     .
     .
  while (GetMessage(&msg, NULL, 0, 0)) {
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }
    return msg.wParam;
}
```

用户如果想使用加速键，就应该使用 LoadAccelerator 函数从资源定义文件中装入加速器表，然后利用 TranslateAccelerator 函数将键盘消息解释为加速键消息。有关加速键的说明，请参见本书的姐妹篇《Windows 高级编程指南》。

用户如使用加速键，其应用主消息循环应具备下面形式：

```
while (GetMessage(&msg, NULL, 0, 0)) {
    if (TranslateAccelerator(hwnd, haccel, &msg) == 0) {
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }
}
return msg.wParam;
```

TranslateAccelerator 函数应出现在标准的 TranslateMessage 函数和 DispatchMessage 函数之前。进一步而言，由于 TranslateAccelerator 函数自动分配加速键消息到相应的窗口程序中，如果 TranslateAccelerator 函数返回非零值，就不应该调用函数 Translate 和函数 DispatchMessage。

1.1.3 消息检验

应用程序可以利用函数 PeekMessage 检验指定消息的消息队列，而无需将它调出队列之外。如果队列中存在这一消息，函数就返回非零值，并允许应用程序访问并处理消

息，无需应用主消息循环。

当应用程序执行较长的操作时，如处理输入和输出，应用程序通常使用 PeekMessage 函数定期检验消息。例如读函数用来检验终止操作的消息。如果消息队列里没有消息，函数 PeekMessage 就放弃，因此当消息不存在时，函数 PeekMessage 就移交控制权给应用程序。

1.1.4 发送消息

函数 SendMessage 和函数 PostMessage 使应用程序传递消息给它们的窗口，或者传递给其它应用程序窗口。函数 PostAppMessage 相对函数 PostMessage 是可变的，因此使用应用模块指针，而非窗口指针传递信息。

函数 PostMessage 通知窗口传递消息意味着将消息置入应用消息队列中。函数 PostMessage 立即返回控制给调用中的应用程序，在从队列中读取消息之前，不会出现任何操作结果。

函数 SendMessage 绕过从应用消息队列，通知窗口直接发送消息到给定的窗口程序，在接收消息的窗口程序处理消息或者调用函数 ReplyMessage 的结果返回控制之前，窗口并不交还控制给调用中的应用程序。

如果应用程序依赖于消息的返回值，它必须通过调用函数 SendMessage 来传输信息。函数 SendMessage 的返回值与处理消息的窗口程序的返回值是一致的。一旦发送完消息之后，函数 PostMessage 立即返回布尔量，指示消息是否成功地置于队列中，但不表示消息是怎样处理的。

1.1.5 避免消息死锁

当应用程序利用函数 SendMessage 处理来自其它应用程序时（或者来自其它应用程序的窗口时），如果它得到控制权就能会生成死锁条件。

通常，一个任务调用函数 SendMessage 发送消息给另外一个任务时，在接收消息的窗口程序返回之前，该任务不会继续运行。当接收消息的任务获取控制权，正在发送消息的任务如果等待函数 SendMessage 返回，并且不继续运行和处理消息，就会因此导致消息死锁。

正处理消息的应用程序不会放弃而产生错误。调用下列任意一个函数将导致应用程序获取控制权：

- DialogBox
- DialogBoxIndirect
- DialogBoxIndirectParam
- DialogBoxParam
- GetMessage
- MessageBox
- PeekMessage
- Yield

在处理消息的过程中调用这些函数之前，窗口程序首先要调用函数 InSendMessage，

以便检查消息是否由函数 SendMessage 从其它应用程序中发送出来。如果函数 InSendMessage 返回非零值，窗口程序在调用任一获取控制权的函数之前应该调用函数 ReplyMessage。

1.1.6 消息函数

消息函数阅读和处理应用消息队列中的消息。下列是消息函数及说明:

函数	说明
CallWindowProc	传递消息给指定窗口程序
DispatchMessage	传递消息给指定窗口的窗口程序
GetMessage	访问应用消息队列中的消息
GetMessageExtraInfo	访问硬件消息
GetMessagePos	当访问到应用消息队列中最后的消息时，返回当时鼠标的位置
GetMessageTime	返回应用消息队列中最后消息的访问时间
GetQueueStatus	返回消息队列中标识应用消息类型的值
hardware__event	置硬件消息到系统队列中
InSendMessage	指出当前窗口程序是否在处理消息，该消息由其它应用程序调用函数 SendMessage 发送出来
PeekMessage	检验应用消息队列，如果消息存在，则在指定范围里返回一消息
PostAppMessage	置消息于应用消息队列中
PostMessage	置消息到与指定窗口有联系的应用消息队列中
PostQuitMessage	传递WM__QUIT消息给应用程序
ReplyMessage	响应从不同任务发出的消息，无需交还控制权
SendMessage	发送消息到一个窗口或一组窗口中
SetMessageQueue	创建一个不同大小的新消息队列
TranslateAccelerator	为菜单命令处理加速键
TranslateMDISysAccel	为多个文档接口（MDI）子窗口处理加速击键
TranslateMessage	把虚拟击键消息解释为字符消息
WaitMessage	给其它应用程序提供控制权
WinMain	执行窗口系统应用程序的入口

有关消息函数的具体说明，请参见本书的姐妹篇《Windows 函数速查手册》。

1.2 建立和管理窗口

这一部分讲述了如何建立、取消、修改窗口及获取窗口消息。

1.2.1 窗口类

窗口类是一组属性，定义窗口的外观及行为。在应用程序生成和使用窗口之前，必须

为这一窗口建立和登记窗口类。应用程序通过填写 WNDCLASS 结构、传递指针给 RegisterClass 函数的结构来登记类。能登记任意数量的窗口类。窗口类一经登记，应用程序就可以建立任意数量的该类的窗口。在窗口类删除或应用程序中止之前，窗口类保持有效。

尽管一个完全的窗口类包含许多元素，但是窗口只要求应用程序提供类名、处理发送到这一类窗口消息的窗口程序地址、标识登记类的应用程序的实例句柄。窗口类的其它元素定义该类窗口的缺省属性，诸如窗口的形状以及窗口菜单的内容。

有三种类型的窗口类：系统全局类、应用全局类、应用程序局部类。这些类型在范围、建立和破坏窗口的时间及方式方面均有不同。

1.2.1.1 系统全局类

启动窗口时就建立了系统全局类。这些类在任何时候对全体应用程序有效。由于窗口系统是为了应用程序所有的，所以建立系统全局类，应用程序不能建立或破坏这些等级。系统全局类包含编辑控制及清单盒子控制类。

1.2.1.2 应用全局类

应用程序（更多时候是动态连接库 DLL）通过给类指定 CS_GLOBALCLASS 方式来建立应用全局类。这些类一经建立，对系统内所有应用程序均有效。通常 DLL 建立应用全局类，那些调用 DLL 的应用程序就能使用类。当建立窗口的应用程序关闭，或者建立窗口的 DLL 卸载时，窗口就破坏应用全局类。因此在建立窗口的应用程序关闭及 DLL 卸载之前，所有应用程序必须利用那一类破坏所有的窗口。可以使用函数 UnregisterClass 消除应用全局类，并释放与之关联的内存空间。

1.2.1.3 应用局部类

应用局部类是应用程序为自己使用而建立的窗口类。这是一种由应用程序建立的更普通的窗口类。可以使用函数 UnregisterClass 消除应用局部类，并释放与之关联的内存空间。

1.2.2 Windows 如何寻找窗口类

当应用程序建立指定类窗口类时，窗口按照下面步骤寻找窗口类：

1.窗口搜索给定名字的局部类。

2.如窗口没有找到这一名字的局部类，它就搜索应用全局类清单。

3.如窗口没有在应用全局类清单中找到该名字，则搜索系统全局类清单。

这一过程适用于所有由应用程序建立的窗口，也包括由窗口为应用程序建立的窗口，如对话框。同时有可能越过系统全局类而不影响其它应用程序。

1.2.3 类属

当应用程序或 DLL 登记窗口类时，窗口根据传递给函数 RegisterClass 的 WNDCLASS 结构的 hInstance 元素确定类属。对窗口 DLL 而言，hInstance 元素必须是 DLL 的实例句柄。当登记类的应用程序关闭及建立类的 DLL 卸载时，该类即已破坏。因此属于该类的窗口需在应用程序关闭及 DLL 卸载之前消除掉。

1.2.4 登记一个窗口类

当窗口登记一个窗口类时，它复制类属性到自己的存储区．当应用程序通过名字确定窗口类时，窗口使用这些内部存储属性．那些开始登记窗口类的应用程序没有必要使结构保持有效．

1.2.5 共享窗口类

一个应用不能与其它应用共享已登记的类．窗口类的一些消息，如窗口程序的地址，是针对指定应用程序的，其它应用程序不能使用．但应用程序之间能共享应用全局类．详细情况参见本手册 1.2.1.2"应用全局类"部分．

虽然一个应用程序不能与其它应用程序共享任一已登记的类，但是，同样应用程序的不同映象可共享一个已登记类．一旦应用程序已经登记一个窗口类，该应用程序的后续映象都可以使用它．这意味着前一个应用程序不应该也不必要登记以前映象已经登记过的窗口类．

1.2.6 预定义窗口类

窗口提供几个预定义的系统全局窗口类．这些类定义执行公共输入任务的专用控制窗口（如让用户控制翻转、打印课文、在名字清单中选择等）．预定义窗口类对所有应用程序有效，可以任意使用以建立任意数量的控制窗口．关于预定义窗口类清单，详见本书的姐妹篇《Windows 函数速查手册》，函数 CreateWindow 说明部分．

1.2.7 一个窗口类的元素

1.2.7.1 窗口类的元素

窗口类的元素确定了该类窗口的性能．登记窗口类的应用程序通过在 WNDCLASS 结构设置合适的成员，并传递该结构给函数 RegisterClass 来指定类的元素．应用程序可使用函数 GetClassInfo 查询指定窗口类消息．窗口类元素可表示如下：

元素	用途
类名	区分已登记的类
窗口程序地址	指向处理发往指定类窗口消息的函数指针
实例句柄	标识登记类的应用程序或DLL
光标类	当光标在某一类窗口中时，定义光标形状
图类	当某一类窗口最小时，定义窗口显示的图的形状
背景画笔类	当窗口被打开或重画时，定义窗口用来填充用户区域的颜色和模式．如果这一参数为空，窗口当接收 WM_ERASEBKGND 消息时需绘上自己的背景
菜单类	指定属于某一类窗口的缺省菜单，类本身不显式定义一个菜单
方式类	确定在窗口移动及缩放之后如何更新窗口、如何处理