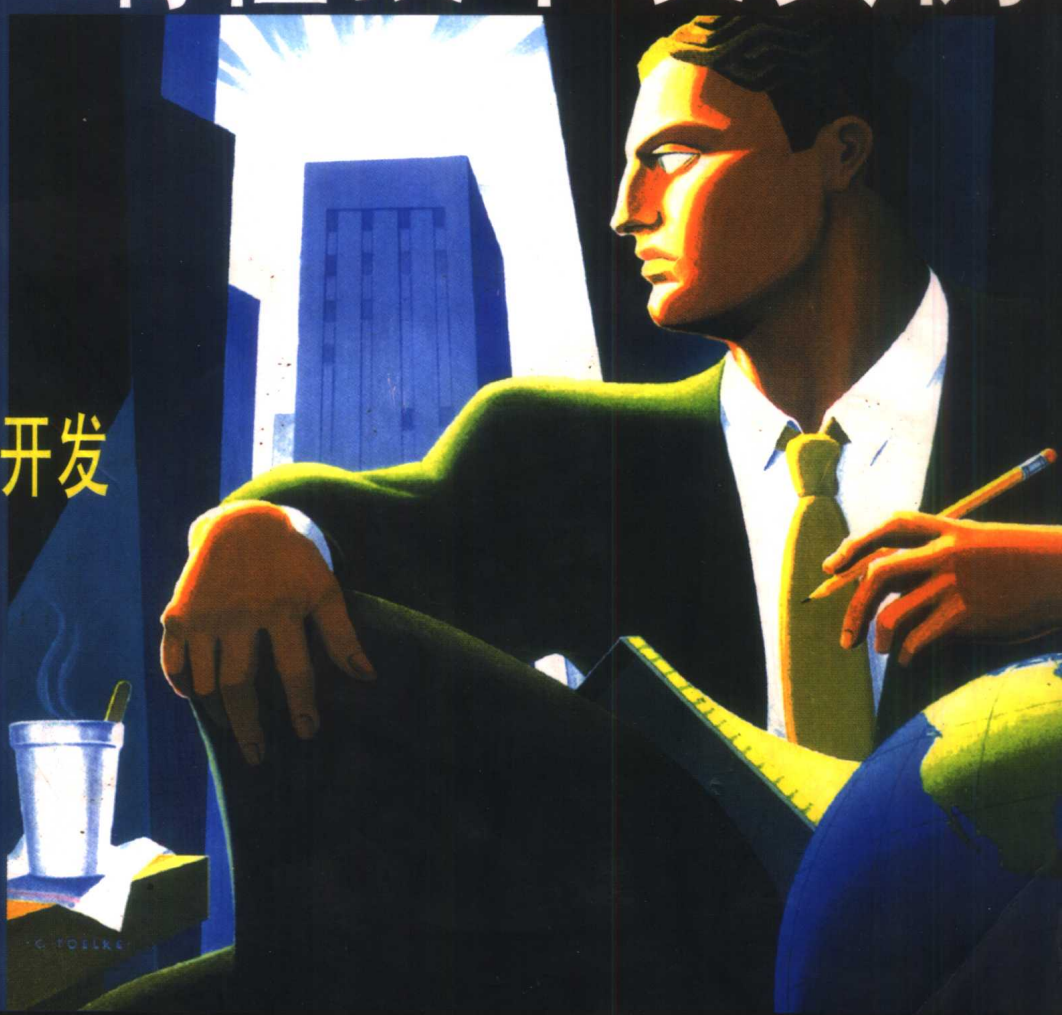
Borland C++ 环境下 Windows 3.1～95 编程技术及实例

王培杰　等编著

计算机软件开发与程序设计系列丛书



机械工业出版社

计算机软件开发与程序设计系列丛书

Borland C++环境下 Windows 3.1～95 编程技术及实例

王培杰　等编著

机 械 工 业 出 版 社

本书为计算机软件开发与程序设计系列丛书之一，书中系统全面地介绍了用C++开发Windows应用程序时遇到的各种问题，包含了开发一个完整的应用程序所需的各种功能，特别是涉及到了Windows 3.1提供的一些高级功能，如DLL、多文档界面、DDE、OLE、钩(Hook)函数，Shell库、安装程序等，这些功能对于开发Windows的高级应用程序是必不可少的。除了这些高级功能之外，也介绍了开发Windows应用程序用到的一些基本功能，如菜单、GDI、对话框、滚动杠、字符输出等。

为进行面向对象的Windows程序设计，我们利用了Borland公司提供的Object Windows类库。本书在介绍Windows功能的同时，也介绍了如何使用此类库，并对比较重要的类进行了剖析，力求使读者对此类库有一个比较深刻的理解。

本书提供了大量的程序实例，这些实例用现在最流行的Borland C++ 3.1开发，并经过调试。这些实例可帮助读者更好地理解面向对象的Windows程序设计，同时，在本书例子中开发的各种类都可作为以后实际开发应用程序的类库。

本书是一个内容比较全面的介绍Windows应用程序开发的参考书，书中内容安排由浅入深，适合于初学者，也可作为Windows应用程序开发人员的参考书。

本书附全部源程序代码盘二张，定价120元，欲购者请与本书责任编辑联系。

图书在版编目(CIP)数据

Borland C++环境下Windows 3.1～95编程技术及实例/王培杰等编著.—北京：机械工业出版社，1997.4

计算机软件开发与程序设计系列丛书

ISBN 7-111-05396-6

Ⅰ.B… Ⅱ.王… Ⅲ.C语言 程序设计 Ⅳ TP316C

中国版本图书馆CIP数据核字(96)第19628号

出 版 人：马九荣(北京市百万庄南街1号 邮政编码100037)

责任编辑：王福俭 蒋 克 版式设计：张世琴

责任校对：姚培新 封面设计：高 科

北京第 外国语学院印刷厂印刷 新华书店北京发行所发行

1997年5月第1版 1997年11月第2次印刷

787mm×1092mm 1/16 57.25印张 1481千字

印数：4001—7000册

定价：96.00元

前　言

Windows 代替 DOS 成为 PC 机的主流操作系统已为大家所公认，利用 Windows 开发应用程序所带来的好处及其强大的功能使得应用软件的开发纷纷转向 Windows 环境，因此关于 Windows 系统下编程的书籍也相应出版了很多，但这些书籍中忽略了软件发展的另一大趋势，就是面向对象的编程，而这种趋势对软件的发展有着更为重要的意义。由于 Windows 系统本身也引入了很多的面向对象的概念，并且 Windows 系统的终极目标是提供一个完全面向对象的操作系统，因此，在 Windows 环境下采用面向对象的编程方法就更为有利，也更有意义。

根据这种情况，作者根据多年编写 Windows 应用程序的经验编写了本书，它详细介绍了在 Windows 环境下如何采用面向对象的编程方法来编写应用程序。编写 Windows 程序之所以入门比较困难，原因是 Windows 采用的是消息驱动的编程方式，这和 DOS 采用的顺序方式有很大的不同，这也是面向对象的编程方法和普通的编程方法的差别，正确地理解概念是编好Windows 程序的关键，因此，本书对这些新的概念都结合实例进行了详细的解说，由浅入深，以使初学者尽快适应 Windows 编程的思路。

本书采用 Borland C++做为编译器，并利用了 Borland C++提供的 Object Windows 类库 (OWL)，并在适当的地方对 OWL 进行了详细的介绍。

本书涉及到了 Windows 程序的各个方面，除了编写 Windows 程序所需的基本内容外，还详细介绍了一些高级技术，如 DDE、OLE、MDI、内存管理、外壳程序及拖放功能等，对 Windows 编程的初学者和 Windows 程序员都有参考价值。

除了介绍 Windows 3.1 的编程外，本书还介绍了 Windows 95 的编程，Windows 95 的编程和 Windows 3.1 的编程差别不大，但其内部机制则有很大的不同，特别是 Windows 95 采用了抢占式多任务和 32 位编程，这大大提高了 Windows 95 应用程序的性能。本书介绍了 Windows 95 编程的主要特点，有了编写 Windows 3.1 程序的经验，这些介绍会使读者顺利进入 Windows 95 的编程。

参加本书编写的作者还有王向明、石建龙、周维国、徐良旺、陈军、刘辉、王楠、曲亮、白伯杨、李新国等。

由于作者水平有限，书中难免存在错误和疏漏之处，恳请读者提出宝贵意见。

作者

1996 年 8 月　于大连

目　录

第 1 章　Windows 简介

Windows 是 Microsoft 公司开发的基于图形的多任务多窗口操作系统，1990 年，Windows 3.0 一推出就以其漂亮的界面、强大的功能、广泛的应用前景受到了广大计算机用户和软件开发商的欢迎，到现在，它的应用越来越普遍，应用领域越来越大，应用软件的增加更加速了Windows 系统的普及。

Windows 编写的应用程序具有一致的外观和命令结构，因此，它比传统的 DOS 应用程序更加易学易用，用户也很容易在不同的应用程序间切换，很容易在不同的应用程序间交换数据。因此，Windows 的应用越来越普及，到目前已经售出了 5000 多万套。国内 Windows 应用程序的普及和国外相比要晚一些，这主要是因为处理汉字的原因，国内开发的软件是为国内服务的，没有汉字的支持，开发的软件也将无法得到广泛的应用，直到北京新天地电脑研究所推出中文之星汉字平台后，才开始普及开来，尤其是今年，可以说是形成了 Windows 热，Microsoft 公司中文 Windows 的推出更是为国内用户使用 Windows 系统提供了一个基础，可以预见，在不久的将来，随着 Windows 应用软件的不断增加，使用 Windows 的用户将超过 DOS，并最终代替 DOS。

1　Windows 的特点

Windows 系统最早由 Microsoft 公司于 1985 年推出，是 Windows 的第一个版本。随后又于 1987 年推出了 Windows2.0，该版本对用户界面做了一些改进。1990 年推出了 Windows3.0，对以前的 Windows 系统做了较大的改进，使其可以支持 Intel 80286 以上芯片的保方式。Windows 3.1 是 1992 年推出的，增加了对 TrueType 字体、多媒体技术、对象链接和嵌入(OLE)技术和通用对话框的支持。

Windows 系统和 DOS 操作系统有很大的不同。首先，从外观上，它是一个图形界面，可以很容易地把图形、图象、文字等结合在一起，使人觉得赏心悦目。第二，它是一个多任务操作系统，可同时进行几项工作，也允许用户在各应用程序间交换数据，这给用户带来了很大的方便。第三，应用程序的设备独立性，Windows 下的应用程序在各种设备上的输出都是相同的，这给应用程序的编制带来了很大的方便。

1.1　一致的用户界面

熟悉 DOS 编程的读者都比较清楚，在 DOS 系统下每个应用程序的界面和开发此程序的程序员有很大关系，每个程序员都可以按自己的意愿和思维方式来构造界面，这当然可以发挥程序设计人员的想象力，使每个软件更有特点，但也带来了问题，首先是界面开发工作量大，界面的开发在应用程序的开发中一般要占 1/3 的工作量，而这部分工作很多是重复性的劳动，减少这部分工作量可以大大加快软件开发的速度。其次，界面的不统一也给用户带来了不方便，因为每一个软件都有自己的使用方法，所以用户对每一个软件都要重新熟悉，花费很多的时间。

和 DOS 系统命令式的操作方式不同，Windows 系统是通过对诸如对话框(Dialog)、图标(Icon)、菜单(Menu)等图形画面和符号的操作来完成的，Windows 系统下所有的应用程序界面都是由标准的界面元素组成，并提供了对这些界面元素的操作函数。每个应用程序都有一个窗口，窗口有菜单，可以通过菜单使程序完成某一功能。用户和程序之间的交互一般通过对话框来

完成，并且对话框的外观和操作也是标准的，图 1-1。

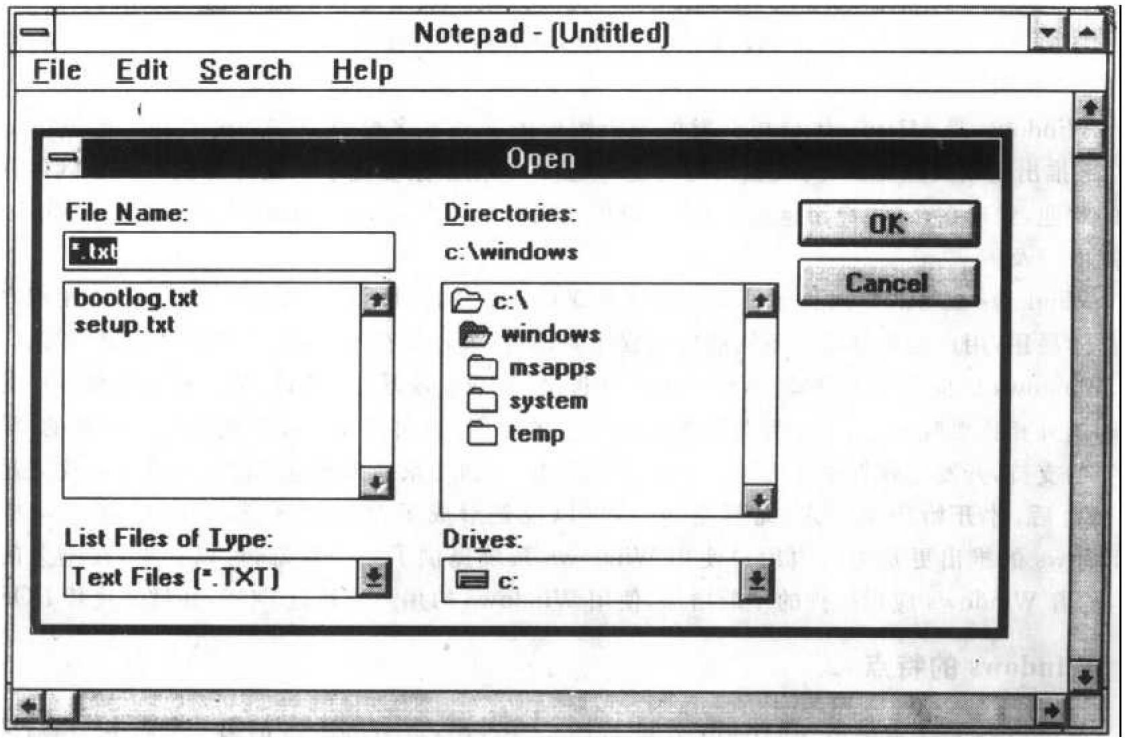


图 1-1 Windows 的标准界面

Windows 的这种标准界面使得用户在学习了一个应用程序的使用之后，也熟悉了其它应用程序的使用，经过很少量的学习就可以使用一个软件。对于编程人员来说，它使得编程人员在界面上的开发工作量减少到最低，因为 Windows 为应用程序的开发提供了丰富的内部子程序，可以很方便地实现弹出式窗口、菜单、滚动杠、对话框、图标等界面元素，并提供了产生这些界面元素的可视化的开发工具，很容易直观地构造出一个界面，而把主要的精力放在程序的功能上，大大节省了程序开发时间，也正是因为界面的标准化才能使 Windows 做到这一点。

1.2 多任务

多任务是指在 Windows 操作环境下，允许用户同时运行几个应用程序，或运行一个应用程序的几个实例。例如，当我们程序编到中间时需要查阅某些东西，如文档、说明或需要拷贝一个文件等等，这些工作在 DOS 系统下就需要结束当前运行的程序，然后才能做其它的工作，但在 Windows 下，你不需要结束当前运行的程序，就让它保持当前的状态，或使当前使用的窗口变为最小，转而去做别的工作，其它的工作完成后，可再回到原来的工作，它还保持原来的状态。也许有人觉得多任务没有必要，但当你使用了 Windows 系统之后，就再也不愿意使用 DOS 了，因为 Windows 的多任务确实给你带来了很大的方便，这说明多任务还是很需要的，尤其是在微机硬件的性能不断提高、内存不断扩大的情况下。DOS 系统中很多内存驻留程序的成功，如 Side-Kick，说明多任务还是很需要的。

图 1-2 是 Windows 下同时运行的几个应用程序，每个应用程序有一个窗口，用户可以随时从一个窗口的操作转到另一个窗口的操作，也可随时改变窗口的大小和位置。

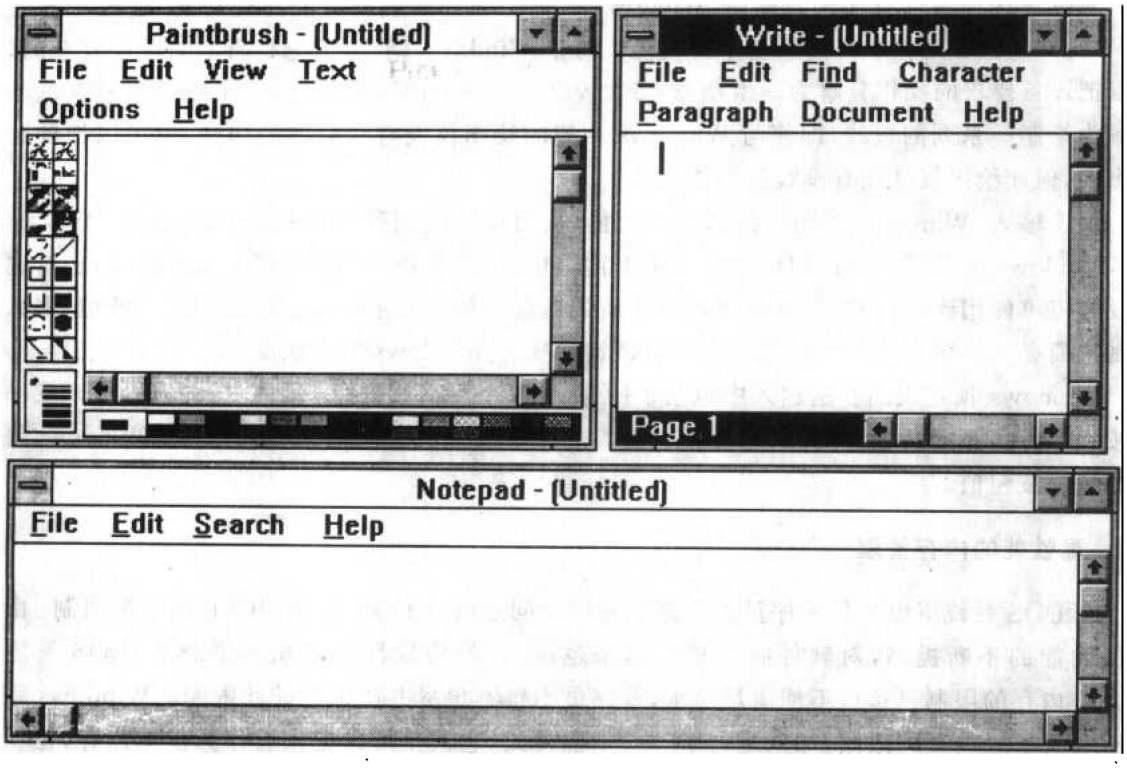


图 1-2　Windows 中的多任务

虽然这几个程序都在运行，但每一时刻只有一个应用程序在使用 CPU，其它程序处于等待状态，使用 CPU 的窗口为活动窗口(Active Windows)，当你用鼠标点一个窗口时，就把它激活了，也就是说它处于使用 CPU 的状态，同时使其它的窗口处于非激活状态，每个时刻只能有一个窗口是处于活动状态的，这是 Windows 的多任务和其它的多任务不同的地方。

1.3　与设备无关性

与设备无关性是指在 Windows 下输出的图形无论是在显示器上还是在打印机上都是一样的。从用户的角度讲，这给用户的使用带来了方便，因为用户在显示器上看到的就是在各种输出设备上(如打印机)输出得到的。从程序员的角度讲，这给编程带了很大方便，因为一个应用程序，不论是 DOS 的还是 Windows 的，都要和各种外设打交道，一般都要用到显示器和打印机。熟悉 DOS 编程的程序员都清楚，在显示器上输出和在打印机上输出是不同的，要为这两种输出编写不同的代码，C 语言库中提供的函数大部分是显示器的函数，要在打印机上输出，程序员要通过设备驱动程序直接操纵打印机，和显示器编程差别很大，并且各厂家生产的打印机需要的驱动程序也不同，因此，如果应用程序支持各种打印机，就需要带很多驱动程序。而 Windows 就不同了，它的编程并不依赖于某种设备，在显示器上输出和在打印机上输出对 Windows 的程序员来说是一样的，这种编程是逻辑上的而不是物理上的，这就使得同样一段程序，既可用于显示器输出，也可用于打印机输出，或用于其它设备的输出。这给编程带来了很大的方便，程序员不用再编写各种各样的驱动程序，应用程序只要与 Windows 打交道而不用管任何具体的设备，不必知道打印机具体的型号，只要发出一个画圆的命令就在相应的设备上画出来。Windows 应用程序用同一种符号和各种设备驱动程序打交道，开发者节省了时间，用户也不必担心每一个新的应用程序是否支持某种设备。

为实现这种与设备无关的特性，Windows 规定了硬件必须具备的几种功能，这些功能是保证 Windows 例程工作所需的最基本的功能。每个 Windows 例程，不论是简单的还是复杂的，都可以把设备要求的操作分解为一组动作。例如，当绘制一个椭圆时，Windows 把绘制椭圆的动作分解为绘制一系列的短线，即使与 Windows 相连的绘图仪没有画椭圆的功能，也可以把椭圆绘制出来，但此绘图仪必须有画线的功能。

关于输入，Windows 说明一组最小的功能，从而保证应用程序只接收合法的预先定义的输入。Windows 定义了一组虚拟键作为合法击键，包含了当前所有键盘上可能出现的键，还有键盘上没有但可能用到的一些键，如果键盘制造商的键盘上包含有不在虚拟键范围之内的键，那么这个制造商必须提供额外的软件把这些"非法"键转换为 Windows 的合法键。

Windows 预定义的合法输入中包括了所有的输入设备，如鼠标。因此，即使有人想开发一种六按钮鼠标也不必担心，只需制造商提供必要的软件来把所有的鼠标输入转换成Windows预定义的鼠标按钮值。

1.4 高性能的内存管理

在 DOS 系统下编程的程序员经常遇到的一个问题就是 DOS 系统 640KB 内存的限制。由于硬件性能的不断提高，对软件的要求也越来越高，这就使软件的体积越来越大，DOS 系统的 640KB 内存的限制已远远不能满足要求，程序员不得不想尽办法来突破此限制。Windows 系统已经取消了对内存的限制，如果运行于 386 增强模式，它甚至可以把硬盘作为虚拟内存，也就是说，如果你的硬盘容量足够，在计算机实际内存比程序所需要的内存小的情况下，照样可以运行程序。

突破 640KB 内存的限制只是 Windows 高性能内存管理的一个方面，除此之外，它还可以动态地移动内存，把零碎的内存块拼合起来形成一个连续的、尺寸较大的内存块，也可以丢弃暂时不用的内存块内的代码或数据，使当前的应用程序获得最大的可用内存。

Windows 高性能内存管理的另一个方面是动态链接机制。对于库函数，相信读者都比较清楚了，这是在编程中经常用到的，实际上大部分的功能由各种库提供，如在 DOS 系统下用 C 编程时，很多工作是调用 C 的库函数，然后用链接器把库函数链接到我们的应用程序中形成可执行的.EXE 文件，实际上是把库程序的代码拷贝到.EXE 文件中，是在程序运行前完成的，这种链接叫静态链接。

在 Windows 系统下编程时情况有所不同，许多 Windows 系统的功能是由被称为 DLL 的动态链接库提供的，它和普通的库不同，不是在链接生成.EXE 文件时把库函数链入应用程序，而是在程序运行时需要的时候才链接，这种链接叫动态链接。所谓动态链接实际上包含三个含义：首先，它是一种内存管理技术，允许按要求从磁盘中把代码调出来，也允许废弃不用的代码，以空出内存为其它应用程序服务。第二，动态链接提供了一种在程序运行时把子例程和程序相链接的方法。在静态链接的情况下，如果库例程更新了版本，应用程序必须重新编译才能使用更新的子例程，而动态链接则不同，如果动态库更新了版本，应用程序可不做任何改动，运行时使用的就是更新的版本。第三，动态链接为在应用程序之间共享代码和数据提供了一套有效的机制。Windows是一个多任务的操作系统，同时可以运行多个应用程序，每个应用程序都调用了Windows的库例程，但不是每个应用程序都有一个库例程的拷贝，而是所有的程序共用一个库例程。例如，如果几个程序同时都要字符输出，都要调用 Windows 的库例程 TextOut，这时内存中只有 TextOut 例程的一个拷贝，几个程序同时使用它，这就使内存的开销达到最小。

Windows 的三个主动态链接库是KRNL 386. EXE、GDI. EXE、USER. EXE，KRNL 386. EXE是Windows的内核，负责程序管理，GDI. EXE负责图形界面的输入输出，USER. EXE 负责用户界面。

2　Windows 用户界面的组成和术语

用户界面是应用程序和用户交互的手段，它直接影响一个应用程序的成功与失败。利用Windows可以很容易地做出使用方便的界面，我们编程的很大一部分工作就是对这些界面的响应，也可以说 Windows 的应用程序开发是围绕着界面元素转的，所以这节将介绍 Windows 的界面元素，使读者对这些界面从外观到术语都有一个概念，在以后的编程中将直接引用这些概念和术语。

2.1　窗口

对用户来说，Windows 的窗口是一个与应用程序无关的矩形区域。它是用户与应用程序之间的可视界面。对于应用程序来说，窗口是该应用程序控制下的屏幕上的一个矩形区域。应用程序创建并控制窗口的所有属性，包括尺寸、形状等。每当运行一个应用程序，一个窗口就被创建；每次用户选中窗口内的一个可选项时，程序就有所响应；关闭一窗口，程序也随之结束。

窗口一般都有边框、标题栏和控制菜单框，另外还可能有菜单栏、滚动杠、消息栏、状态杠和控制栏，如图 2-3 所示。

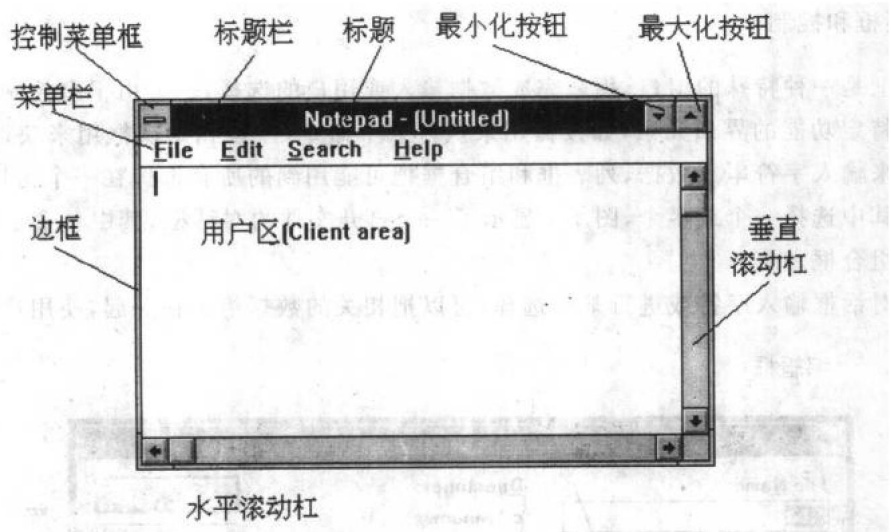


图 1-3　Windows 窗口的基本组成

边框定义窗口的边界，当多个窗口覆盖时，用于相互区别。所有的窗口都有边框，除非该窗口被最大化，填满整个屏幕（此时窗口边框与屏幕边重叠）。大部分窗口都有一改变大小的边框。当光标在窗口的边框上时，变成改变窗口大小的光标形状。用户可以用鼠标拖动边框改变窗口的大小。拖动过程中，光标下是一灰色虚框；鼠标按钮释放后，窗口在新的位置上重画。拖动窗口的某一个角（两相邻边框的交点），则同时改变窗口的宽和高，在水平框线上拖动改变窗口高，在垂直框线上拖动则改变窗口的宽。对话框的边框是不能重置大小的。

标题栏用来显示一个窗口的标题，它包含一标题、系统菜单框、最小化铵钮和最大化按钮（如果窗口不是最大化的）或恢复按钮（如果当前窗口是最大化的）。如果窗口不是最大化的，那么用

户可以用鼠标拖动标题栏使窗口移动到新的位置。在拖动过程中，光标下是一阴影框窗口。鼠标按钮释放后，窗口在新的位置重画。

标题是标识应用程序窗口的名字，当用户在多个应用程序下工作时，标识特别重要。至少标题应包含一首字母大写的应用程序名(如"Write")。标题中也可包含其它字符。例如，应用程序在一数据文件上工作，应用程序名后应跟当前目录和当前工作文件名(用大写字母表示并包含扩展名)。如果当前没有活动文件，标题中应含有替代名，如(Untitled)等。这个替代名应是大小写混合，提示用户这不是一个法定的文件名。

控制菜单包括改变窗口的大小、位置和关闭窗口等命令。每个可移动的窗口(应用程序、文档和对话框)标题栏的左端都有一个控制(Control)菜单。Control 菜单为当前窗口提供了键盘控制命令。

最大化、最小化和恢复按钮是控制菜单中相应命令的图形表示。点按最小化按钮使窗口最小化(一般用图标表示)并隐藏所有相关窗口。点按最大化按钮使窗口变成最大(一般填满整个屏幕)。当窗口变成最大时，最大化按钮被恢复按钮所替代；此时点按恢复按钮或选择控制菜单中的 Restore 命令后，窗口将恢复到它的先前的尺寸。固定尺寸窗口没有最大化按钮。

菜单栏是用来访问应用程序中命令的，一般放在应用程序标题栏的下面。应用程序窗口的菜单栏中包含菜单标题，菜单标题中是表示命令的项。

滚动杠分为水平滚动杠和垂直滚动杠。它的作用是当整个数据内容在窗口中不能完全显示出来时，可以通过滚动杠操作来移动数据，使得能够观察到数据的隐藏部分。

2.2 对话框和控制

对话框是一种特殊的窗口，用来完成数据输入或用户的选择，一般由很多控制组成。控制是完成某种特定功能的界面元素，如按钮用来执行某个命令，按钮和确认框用来获得用户的选择，编辑框用来输入字符串或数据，列表框和组合框把可能用到的所有选择在一个窗口中列出来，用户可以在其中选择一个或多个，图 1-4 显示了一个打开文件的对话框，其中包含了按钮、编辑框、列表框和组合框。

使用对话框输入字符或进行某种选择，可以把相关的数据组织在一起，使用户觉得使用起来

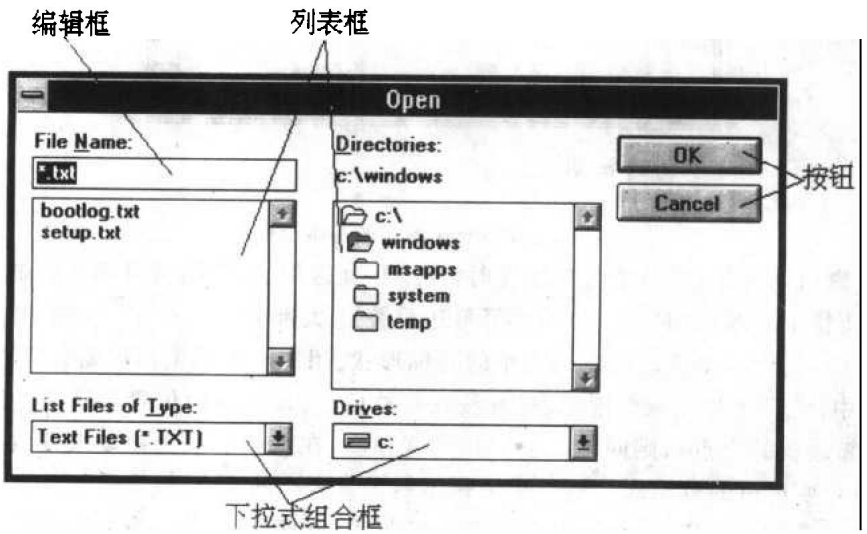


图 1-4 对话框及其控制

很方便，并且由于对话框及其控制的图象很形象，也使用户使用起来不觉得枯燥。

3　Windows 的资源

资源是一种特殊的数据，但它并不和程序的数据段在一起，通常它是独立设计的并单独编译，最后由链接程序把资源数据和程序代码链接在一起。如上面介绍的对话框和控制就是一种资源，经常用到的其它资源还包括图标、光标、插入符(caret)、位图、字体、画笔和刷子等。

当 Windows 把一个程序装入内存准备执行时，通常把所有的资源留在磁盘上。例如当用户首次查看应用程序的 About 对话框时，该应用程序必须先访问磁盘，把该信息从程序的.EXE 文件中拷贝到内存里，然后才能显示 About 对话框。

通常情况下，应用程序把自己的资源定义为“只读”和“可丢弃的”，这就使得 Windows 在需要更多的内存时可以丢弃资源，一旦再次需要，只需从磁盘上把数据重新装入内存。

3.1　图标

图标是一个小的图象，用来代表某个操作或作为一个应用程序的标志，如我们使用 Paint Brush 时，把它缩到最小，它就变为一个彩色的调色板，代表此应用程序还在运行，用鼠标双点此图标又恢复原来的状态。图标以直观的方式告诉用户某些信息，即节省了空间，取得了用文字描述无法取得的效果。如一个警告信息，加上一个感叹号不仅引起用户的注意也使警告的气氛很浓，这是单纯用文字无法取得的效果。图标可以用 Microsoft 的 SDK 或 BorlandC++的 Workshop 来制作，制作很容易，并可用函数很容易地显示在需要的地方。

3.2　光标

在 DOS 系统下，光标是指显示字符位置的闪烁的短横线，但在 Windows 中光标是指用来表示鼠标位置的各种图形，如箭头、十字线、I 字线、沙漏型等，一般每种鼠标都代表一定的含义，如十字线一般用于图形系统中的点的定位，I 字型一般用于字符输入，沙漏型表示程序正在执行某种操作要求用户等待等，当然，也可以自己定义一个任意形状的光标，也可以赋给它某个特定的含义。和图标一样，用 SDK 或 Workshop 很容易制作一个光标。

3.3　插入符

Windows 的插入符和 DOS 系统下的光标相似，用来表示当前操作字符的位置，它始终是闪烁的，所以很容易和光标区分开来。一般插入符作为键盘输入的标记，而光标作为鼠标输入的标记。

3.4　位图

一个位图可视为存入内存的显示屏一部分的瞬态图。当一个应用程序需要快速显示一幅图象时就要用到位图，因为该对象是从内存中直接传送的，所以要比用代码生成图象的速度快得多。位图有两种用途，一是在显示器上画图，位图的另一种用途是创建刷子。使用位图也有两个缺点，一是位图可能占用很大的内存，另一个是位图的图象是不能由程序随意改变的。

3.5　字体

字体定义了一种大小相同、风格统一的字符全集。Windows 提供了很多的标准字体供选择，