

Windows

应用软件开发方法、 技巧与实例

方旭明 王倩 编著



- 中文 Windows 3. x 环境
- Visual C/C++ 编程
- 位图菜单设计技巧
- 实用窗口技术
- 图形编辑集萃
- 程序可直接引用

西 南 交 通 大 学 出 版 社

Windows

应用软件开发方法、 技巧与实例

方旭明 王 倩 编著

西南交通大学出版社

内 容 提 要

本书深入浅出地介绍了 Windows 应用软件开发的方法和技巧。这本书不是 Windows 的“短平快”，也不是 Windows 程序设计基本原理的简单罗列，而是围绕着一个应用软件开发过程中所遇到的若干常见问题而展开的讨论，如窗口设计、菜单设计、资源管理和信息的输入与输出等等。其范围涉及到 Windows 程序设计的大部分内容。读者在初步掌握 Windows 程序设计方法的基础上，可以先抛开一般书上介绍的冗长概念，尝试着借助本书给自己的应用软件搭出一个框架，然后再去精雕细琢。书中的很多示例取自于作者开发的通用图形编辑软件——GEditor。读者通过循序渐进的学习，可从逐步掌握原理过渡到随心所欲地开发自己的应用软件。本书的所有程序均用 Microsoft 公司新近推出的 Visual C/C++ 1.0 编写（也适用于 Microsoft C/C++ 7.0 和 Borland C/C++ 3.1），在中文 Windows 3.1 环境下运行。为了便于读者使用本书，书中所有程序示例被集中存放在一张 3.5 英寸的软盘中。相信读者通过本书的学习会很快掌握 Windows 的编程方法和技巧，并可直接引用书中的程序段。本书会成为 Windows 软件开发一本难得的参考书。

Windows 应用软件开发方法、技巧与实例

方旭明 王倩 编著

*

西南交通大学出版社出版发行

（成都 二环路北一段 610031）

新华书店经销

郫县印刷厂印刷

*

开本：787×1092 1/16 印张：17.25

字数：430 千字 印数：1—5000 册

1996 年 3 月第 1 版 1996 年 3 月第 1 次印刷

ISBN 7—81022—887—0/T·165

定价：20.00 元

前 言

Windows 环境下的软件开发已经开始从系统软件向应用软件挺进,近两年各种软件如雨后春笋般地出现。很多软件工作者都欲踏入 Windows 这块领地,想去尝尝 Windows 软件开发的酸甜苦辣。有经验的同行都有这样的感觉:学习 Windows 程序设计不是一项一蹴而就的工作,需要先选上几本好的参考书,然后静下心来慢慢地钻研。市面上的有关资料是越来越多,只是结合实际应用的书太少。有鉴于此,作者呕心沥血,将自己在开发 Windows 应用软件过程中的经验汇集成书,奉献给广大的读者。

本书的一个主要特色就是自始至终围绕着一个图形编辑软件——GEditor 的设计过程,去讨论 Windows 软件开发的诸多概念、方法和技巧等。GEditor 原本是为某监控系统而设计的配套软件,用 Visual C 编写。由于它具有很强的独立性和通用性,因此又可以作为通用软件使用。又因为该软件的设计内容很具有代表性,所以对软件设计者来说,这是一个难得的实例。为了能够最充分地利用本书,建议本书的读者在使用本书之前最好先掌握少许 Windows 程序设计的基础知识。

Windows 是一个复杂的操作系统,其内容浩繁,本书无意去包罗万象,只是想借助本书帮助读者迅速掌握其基础,然后渐入佳境。这也是本书的特色。

本书第一章的第一节至第四节、第三章的第六节至第七节、第四章的第一节、第四节、第五节、第五章的第五节、第六章、第七章除第五节和第八章均由方旭明编写,其余由王倩编写,全书由方旭明统稿。

在编写本书的过程中始终得到西南交通大学出版社特别是张雪同志的关心和支持,作者在此表示由衷的敬意。

限于编著者的水平,对书中错误和欠妥之处,恳请读者批评指正。

作 者

1995年9月于成都

目 录

第一章 Windows——程序设计新境界	1
1.1 窗口——全新的用户界面	1
1.2 面向对象的程序设计	2
1.3 多任务操作的特点	3
1.4 与设备无关的图形界面	4
1.5 通用图形编辑软件 GEditor 简介	4
第二章 Windows 应用软件开发环境与开发语言	7
2.1 Windows 开发环境	7
2.1.1 硬件开发环境	7
2.1.2 软件开发环境	7
2.2 最佳开发语言 Visual C/C++	10
2.2.1 Visual C/C++的基本构成	10
2.2.2 Visual C/C++和 SDK 开发系统	12
2.2.3 Visual C/C++和 MFC 开发系统	13
第三章 灵活多变的窗口、子窗口控制和对话框的设计	15
3.1 主窗口设计	15
3.2 子窗口和弹出式窗口设计	23
3.3 子窗口控制设计	23
3.4 对话框设计的一般方法	31
3.5 模式对话框和无模式对话框的设计	35
3.6 Windows 通用对话框简介	44
3.6.1 文件操作对话框	44
3.6.2 字体操作对话框	47

3.6.3 颜色操作对话框	48
3.7 两种常用子窗口设计实例	50
3.7.1 GEditor 时钟子窗口	50
3.7.2 GEditor 光标坐标子窗口	56
第四章 形象直观的菜单设计	62
4.1 文本菜单的制作方法	62
4.2 菜单的在线修改	70
4.3 浮动式菜单设计	80
4.4 位图菜单设计	88
4.4.1 用位图取代文本字符	89
4.4.2 按钮型位图菜单设计	95
4.5 GEditor 位图按钮菜单的设计实例	102
第五章 内存的有效使用	105
5.1 局部堆和全局堆	105
5.2 内存分配及锁定	106
5.3 内存使用、解锁与释放	107
5.4 系统模式与动态内存管理	115
5.5 GEditor 的内存管理	116
5.5.1 GEditor 的链表数据结构	117
5.5.2 GEditor 的内存动态管理	118
第六章 鼠标与键盘使用的合理分配	123
6.1 鼠标消息的处理	123
6.1.1 鼠标消息处理示例	124
6.1.2 Shift 键与 Ctrl 键的处理	124
6.2 键盘消息的处理	125
6.3 鼠标与键盘的最佳使用场合	136
6.4 用键盘模拟鼠标	137

6.5 GEditor 鼠标与键盘管理	143
第七章 图形绘制方法与技巧	149
7.1 图形的绘制	149
7.1.1 颜色的选取	149
7.1.2 点的绘制	150
7.1.3 线的绘制	151
7.1.4 填充图形的绘制	155
7.1.5 区域的处理	177
7.2 字体及文字输出	180
7.2.1 字体和字形	180
7.2.2 文字的输出生	181
7.3 GEditor 图形输出的若干技巧	183
7.3.1 图形的绘制	219
7.3.2 图形块移动	220
7.3.3 图形块拷贝	222
7.3.4 图形块删除	223
7.4 不等宽字体字符的编辑	223
7.4.1 字符宽度的度量	224
7.4.2 编程实例	224
7.5 打印输出技术	233
7.5.1 彩色位图的打印	234
7.5.2 彩色位图的分带打印	236
7.5.3 GEditor 图形的打印	239
第八章 用户文件的管理	249
8.1 文件操作通用对话框的应用	249
8.2 文件的读写操作	252
8.3 GEditor 的文件管理	257
参考文献	268

第一章 Windows——程序设计新境界

Windows 是 Microsoft 公司开发的基于 MS-DOS 操作系统的一个具有多任务能力和图形化用户界面的窗口式操作环境,俗称 Windows 操作系统或 Windows 开发平台。Microsoft 公司声言:Windows 操作系统是一个改变用户使用个人计算机方式的软件。

几年前在中国,对众多的程序员来说,Windows 仍是一个陌生的领域,如今在 Microsoft 公司强大的攻势下,Windows 产品真可谓遍地开花。其中包括:操作系统 Windows 3.1 中文版,高级操作系统 Windows NT 3.1,字处理软件 Word for Windows 5.0 中文版,电子表格和商业图形 Excel for Windows 5.0 中文版,简报制作软件 Power Point 4.0,数据库语言 Fox-Pro for Windows 2.5,编程语言 Visual C/C++ 和 Visual Basic,Windows 环境下的各种应用软件等等。Windows 操作系统本身也在不断推陈出新,1995 年 8 月 Windows 95 又粉墨登场。这些软件都有一致的外部特征和命令结构,用户只要学会其中一个软件,对任意一个新软件都不会陌生。广大程序员和用户已充分领略到它的无穷魅力。用过 Windows 产品的用户在选择今后的软件产品时会毫不犹豫地选择 Windows 平台下产品;而程序员也会在这种形势的鼓舞下,转向在 Windows 平台下开发软件产品。

与传统的 DOS 相比,Windows 的魅力究竟在哪里?总的来说,它具有:① 一种全新的优美、统一的用户界面;② 与设备无关的图形接口;③ 面向对象的程序设计特点;④ 多任务操作;⑤ 功能强大的内存管理;⑥ 动态数据交换。

本章介绍其中的主要特点,并说明这些特性在应用程序中所起的作用。最后简要介绍一下贯穿全书的一个实例——通用图形编辑软件 GEditor。

1.1 窗口——全新的用户界面

Windows 的设计思想完全突破了传统 DOS 系统的局限性,不再有屏幕的文本方式和图形方式之分,系统的全部信息都在统一的图形方式下交换,这种图形用户界面(GUI)都具有基本一致的外观,即占据一个窗口——屏幕上的一个矩形区域,它包含一个标题栏和其它一些控制窗口的部件。Windows 下一个典型的界面如图 1.1 所示。

Windows 的用户有两类:一类是操作员,对他们来说,完全不用再去两眼茫然地盯着屏幕苦思冥想 DOS 命令,屏幕上提供了足够的图形和文字信息告诉用户可以做什么和怎么做。另一类用户是程序员,他们再不用花费很多时间和精力去设计下拉式菜单和子窗口等,菜单、子窗口和对话框设计等是举手之劳,更重要的是用户界面的设计只需要占据开发者很少的时间。

Windows 的界面是交互式的界面,任何操作都可以通过激活菜单项或图标(窗口中的小

图形块)来实现,有时候菜单又会激活一个对话框或图标又会激活一个菜单,向用户提供更多的操作信息。用户可以与屏幕上的对象直接交流。Windows 除了保留键盘接口外,又增加了鼠标(包括轨迹球、光笔、PurePoint 及其它定位设备等)接口。尽管大多数情况下,使用鼠标更方便,Windows 还是同时支持键盘和鼠标两种操作。

Windows 屏幕不仅可以保留多个应用程序的窗口,还支持应用程序在一个用户区内创建多个文档窗口(MDI),分别用来显示和处理不同的文档。

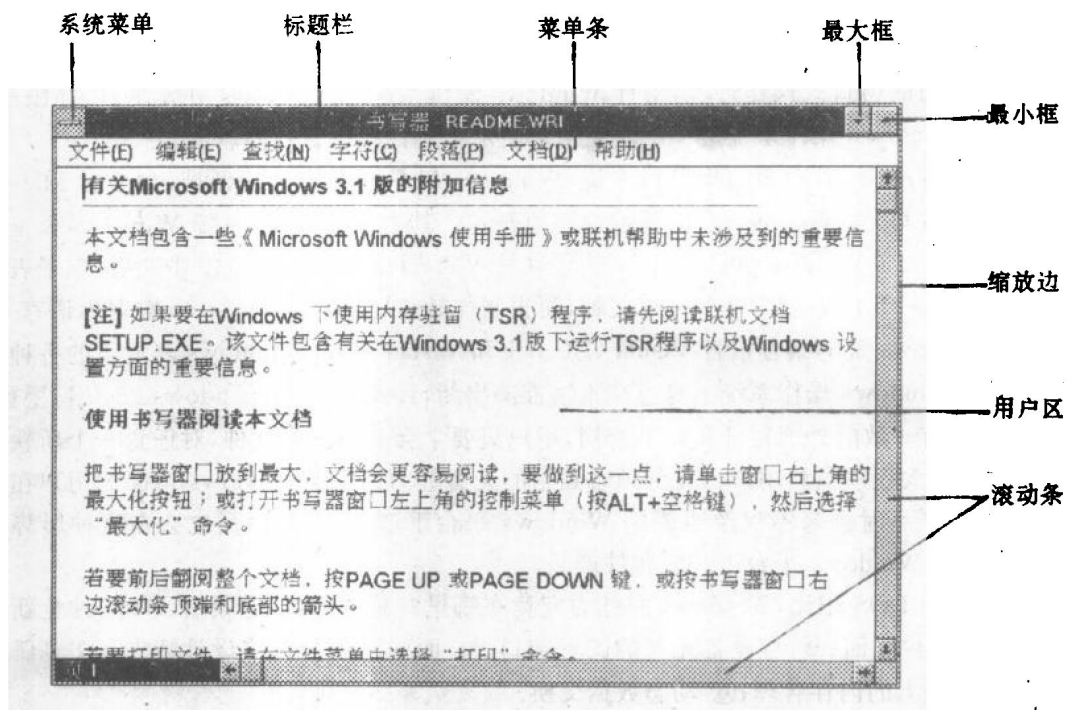


图 1.1 典型的 Windows 窗口界面

1.2 面向对象的程序设计

Windows 的程序设计是一种面向对象的程序设计。Windows 具有的对象很多,如窗口、对话框和菜单等。而使用得最多的就是窗口对象。

窗口就是图 1.1 所示的矩形区域,它可以通过键盘或鼠标接受输入,并将图形输出到用户区域上。一个标准的窗口通常包括标题栏、菜单、滚动杆等,要在用户程序中创建一个窗口,即要建立一个窗口对象。

对话框则是一种特殊的窗口,这种窗口通常包括几个其它窗口,称为子窗口。这些子窗口有多种形式,有压入按钮、单选按钮、组合框、列表框和滚动杆等。这些子窗口又称为子窗口控制或控制。用户将这些子窗口看成是控制对象,可通过按一下按钮或滚动一下滚动杆与这些对象直接交流。对于程序员来说,程序员通过窗口函数(对象中的函数)和窗口消息(对象中的数据)来实现对对象的处理。

菜单也是一种对象。它已经被现代软件普遍采纳。用户通过对菜单的操作可以获得一个应用程序所有操作的分类,直接进入所要进行的操作。

总之,由于 Windows 采用了面向对象的程序设计,使得用户的程序设计方法更科学、更先进。

1.3 多任务操作的特点

Windows 又是一个多任务的系统,它可以同时运行和显示多个应用程序,每个应用程序占据一个矩形窗口,如图 1.2 所示。用户可以在屏幕上移动、在不同的程序间切换,以及在任意两个程序间交换数据。但从严格意义上来说,由于 Windows 是在单任务的 DOS 环境下运行,使用的是一种非抢先方式(nonpreemptive)多任务调度算法,即 Windows 的每个程序只是一个驻留内存的弹出式程序,因此,它不是真正意义上的多任务系统(Windows 95 有了重大改进,用户可运行具备抢先式多任务、多线程和内存保护能力的 32 位应用程序)。

Windows 提供了一种称为“剪贴板”(Clipboard)的数据缓冲区,用于应用程序间的数据交换,此外,还提供了更高级的数据交换形式——动态数据交换(DDE)和动态链接库(DLL)。有关 Windows 的数据交换,可参见书末所附的参考文献。

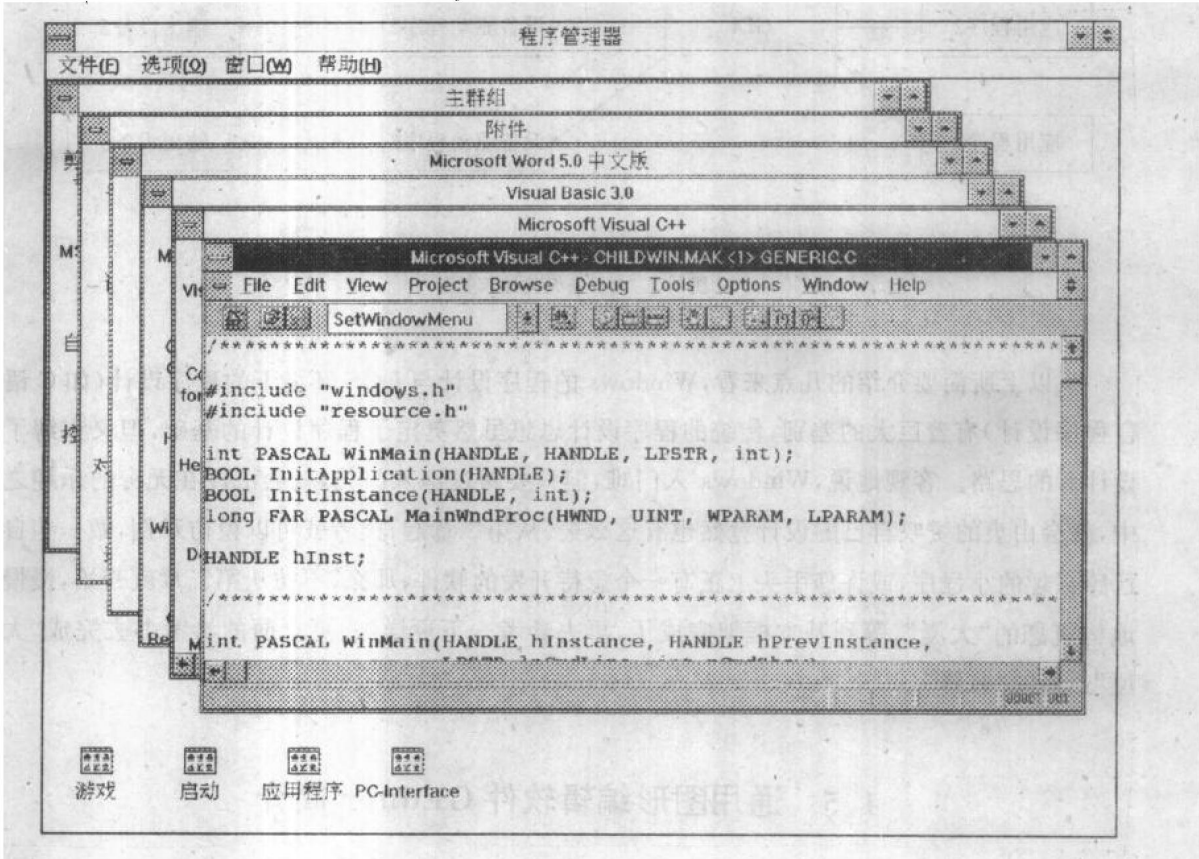


图 1.2 在 Windows 下运行多个程序

1.4 与设备无关的图形界面

Windows 提供了丰富的图形界面,这些图形不是简单的只和显示器及打印机这类输出设备相对应,而是与输出设备无关。由于 Windows 提供了这种与输出设备的无关性,因而用户在某一环境下开发的图形程序可以在不同的输出设备上输出同样的图形。Windows 图形的绘制是借助一种图形程序设计语言(GDI—Graphic Device Interface)来完成的。

Windows 是如何实现这种设备无关性的呢?它采用的是设备驱动程序。只要为 Windows 安装上驱动程序,就可以把 GDI 所设计的图形转换成某种外设所对应的图形。转换示意图如图 1.3 所示。要用户在应用程序中去设计这样的驱动程序是一件不容易的事情,而且也是一项重复劳动。因此,Windows 的系统设计者鼓励外设(如显示卡、打印机、绘图仪、键盘和鼠标等)的制造厂家自行设计外设驱动程序。实际上,为了促销生产厂家也乐意这样做。

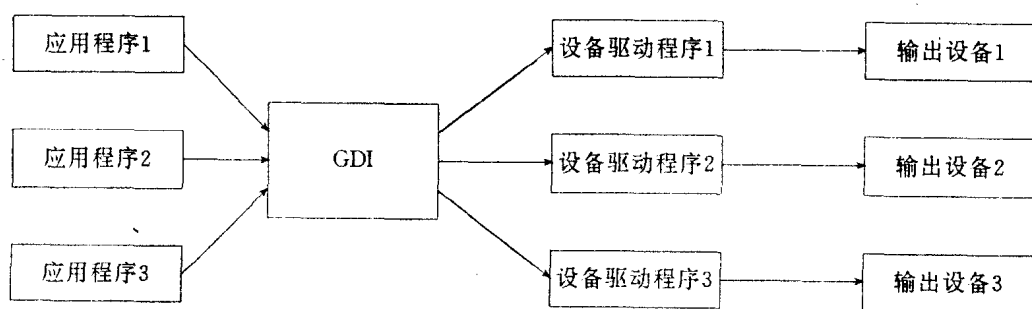


图 1.3 与设备无关性示意图

从以上所简要介绍的几点来看,Windows 的程序设计与 DOS 环境下的程序设计(如 C 语言程序设计)有着巨大的差别,传统的程序设计思想虽然奠定了程序设计的基础,但又束缚了设计者的思路。客观地说,Windows 入门难,但只要跨过这入门关,就会沉浸在无穷的乐趣之中,就会由衷的赞叹自己的设计竟然也有这么美。从第二章起,读者就可以模仿示例,做一些自己感兴趣的小程序,或许您手头上正有一个急待开发的软件,那么,不妨从第二章就开始,慢慢地构筑您的“大厦”,等到基本框架完成了,再去参考一下更详细、更全面的参考书去完成“大厦”最后的“装修”。

1.5 通用图形编辑软件 GEditor 简介

在 Windows 应用程序设计时常要涉及到许多用各种图形表示的画面作为用户界面,用户

可对画面进行修改、增删、拷贝、移动、存储,还可对画面对象进行操作。Windows 下有一通用绘图软件 PaintBrush,但它的画面是以位图方式存储,占空间大,画面对象不易操作。

这里介绍的通用图形编辑软件(GEditor)克服了位图型编辑软件的缺陷,设计了灵活的用户界面如工具箱、图符库、线型框、调色板等丰富的工具,并以下拉式菜单提供:① 画面的剪切、复制、粘贴和打印;② 图形文件的打开、保存和另存;③ 图形查看和各子窗口的开关;④ 光标步距、图符放大系数选择;⑤ 字体及颜色选择等。画面以面向图形对象的链表结构方式存储,比以位图方式存储节约空间数十倍。由于 Windows 系统具有图形界面丰富、多任务操作和动态数据交换等特点,因而 GEditor 选用了 Windows 作为运行环境。图 1.4 表示 GEditor 的界面,图 1.5 是 GEditor 编辑画面示例。

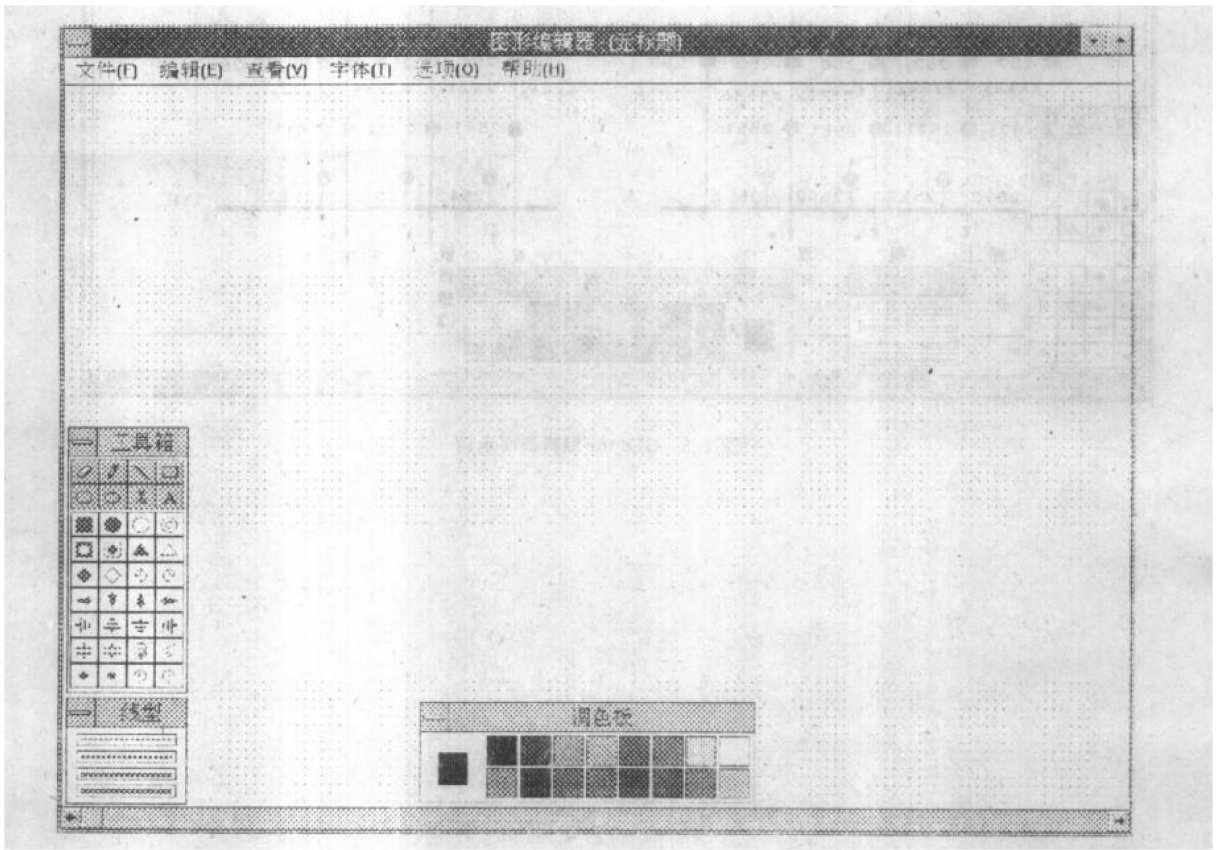


图 1.4 GEditor 界面

GEditor 虽然只是一个通用图形编辑软件,但它的设计几乎涉及到 Windows 程序设计的所有方面,通过 GEditor 的设计我们可以掌握 Windows 程序设计的诸方面方法和技巧,对 Windows 程序设计者颇有启迪。

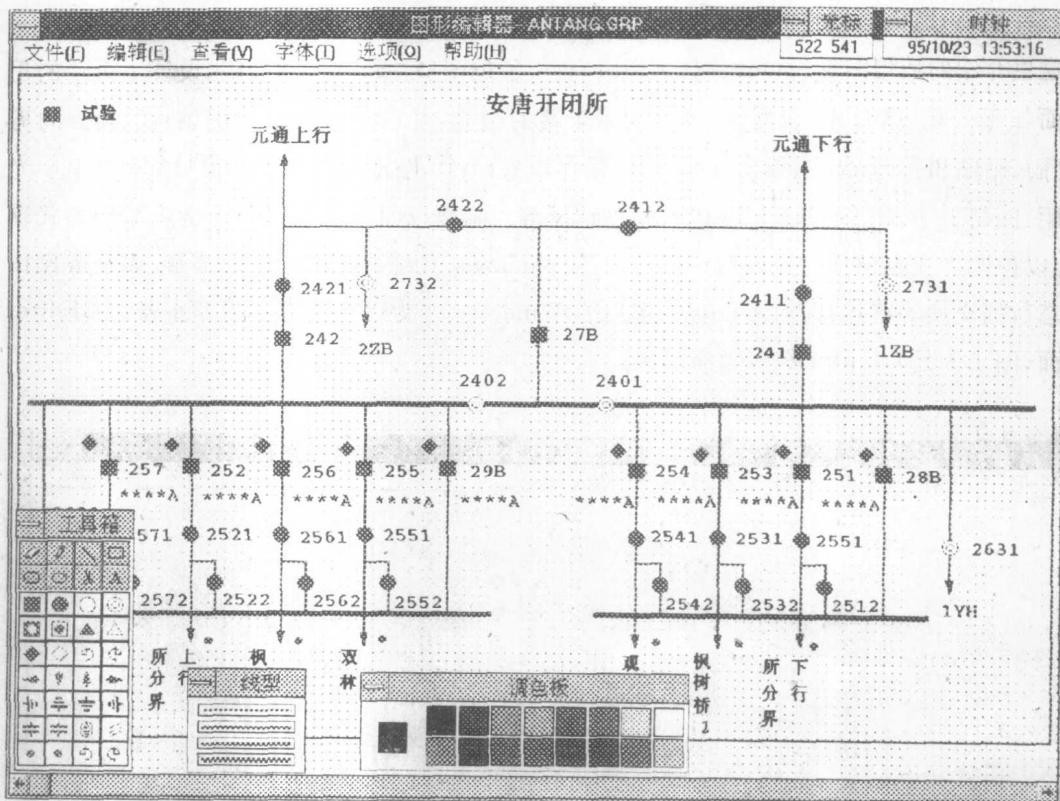


图 1.5 GEditor 编辑画面示例

第二章 Windows 应用软件 开发环境与开发语言

用户在首次着手开发 Windows 应用软件之前,首先要在自己的个人计算机系统上建立一个 Windows 开发环境,选择安装一种最佳的开发语言。本章简要介绍 Windows 系统对用户个人计算机硬件系统的基本要求、推荐使用的开发环境和开发语言。

2.1 Windows 开发环境

Windows 的开发环境分为硬件开发环境和软件开发环境两个方面。这是用户对应用软件进行高效开发的前提。

2.1.1 硬件开发环境

开发 Windows 应用软件需要一定的软硬件环境。为了能使 Windows 应用软件的性能得到充分体现,并能在硬件上配置较好的软件开发环境,推荐使用的硬件环境为:

- (1) IBM PC 或者百分之百兼容机。
- (2) 一个 VGA 或 SVGA 彩色显示器。
- (3) Intel 80386 或更高处理器。
- (4) 640K 常规内存,加上 4M 扩展内存(推荐 8M)。
- (5) 120M 以上空闲硬盘空间。
- (6) 一个 1.44 M 3.5" 软驱或 1.2 M 5.25" 软驱。
- (7) 一个 Windows 支持的鼠标器,虽然它不是必要的,但安装了鼠标器就能更充分地应用易于使用的 Windows 图形界面。
- (8) 如果要用 Windows 打印,需要配置一台 Windows 支持的打印机。

2.1.2 软件开发环境

除硬件开发环境外,开发 Windows 应用软件还需一定的软件环境:

- (1) MS-DOS 3.3 版或以上版本。

(2) Windows 3.1 操作系统。对于中文用户可选用 Windows 3.1 中文版。

(3) Windows 下软件开发工具。

MS Windows 是一个在 DOS 之上的多任务操作系统,它提供了用户图形界面,高级应用程序接口,实现了动态数据交换(DDE),动态数据链接(DDL),自动内存管理等功能,突破了 MS-DOS 的局限性。同时,Windows 也是一个完善的软件开发平台,在 Windows 平台上已推出多种应用软件编译器及开发工具,为 Windows 下应用软件设计提供了方便。以下是几种新出现的基于 Intel 80x86 系统(包括 Pentium)和 Windows 3.x 操作系统的 C/C++ 编译器及开发工具。

1. Microsoft Visual C/C++ 1.0/1.51

Microsoft Visual C/C++ 1.0/1.51 是 Microsoft 公司新一代 C/C++ 产品。它与以往的 MS C/C++ 开发工具相比又上了一个新台阶。

Visual C/C++ 是一个基于 Windows 的交互式开发环境,它包括:高集成度与 Windows 环境全兼容的 Visual 工作平台;多种开发工具如 AppStudio, AppWizard, ClassWizard 等;新的基本类库 MFC 2.0。Visual C/C++ 有标准版和专业版两个版本。标准版是 Microsoft Quick C 的升级产品,专业版除具有标准版的所有功能外还具有优化编译器,基于 MS-DOS 的应用程序开发及完全的 Windows 3.1 SDK。Visual C/C++ 已成为目前越来越流行的 Windows 下软件开发工具。这也是本书中所采用的开发工具。目前,Visual C/C++ 已升级到支持 32 位操作的 2.0 版。

2. Borland C++ 3.1

Borland C++ 3.1 包括两个 IDE:一个基于 DOS,一个基于 Windows。基于 DOS 的 IDE 可产生 DOS 和 Windows 的应用程序;基于 Windows 的 IDE 可生成 Windows 可执行文件和 DLL 文件。

尽管 Borland C++ 的环境不包括等同于 Microsoft 的 AppWizard、ClassWizard 的工具,它也包含了足够多的 Windows 工具,如 Borland 资源工作间(Resource workshop)、帮助编辑器、资源编译器等。Borland C++ 3.1 含有 DOS 和 Windows 的类库:基于 DOS 的 Turbo Version 和基于 Windows 的 Object Windows Library (OWL)。Turbo Version 可以生成基于字符窗口系统下的 DOS 应用程序;OWL 是针对 Windows 环境的,可生成 Windows 应用程序。随着 DOS 和 Windows 开发工具的发展,Borland C++ 已成为一个出色的开发系统。目前,Borland C/C++ 已升级到支持 32 位操作的 4.5 版。

3. Symantec C/C++ 6.0

Symantec C/C++ 6.0 是 Zortech C++ 的更高版本。Symantec C/C++ 包含许多工具,它提供了一个固化的类库——一个 MFC 2.0 注册版本,来为 Windows 开发提供框架。

Symantec C/C++ 的 IDE 很有特色,与通常的 MDI(多文档界面)方法不同,它生成具有

松散联系的多个窗口和工具条,组织在工作平台内的一个环境。每个工作平台(可以同时具有多个工作平台)提供一个工程视窗(如:一个工作平台为用户编辑和编译,另一个包括所有的调试工具)。Symantec 的 IDE 组合了多个工具模板,可以为它们定位、定大小、定做工作平台。这些模板中的图标用拖拉方式进行操作。Symantec C/C++ 的 Visual Programmer 与 Visual C/C++ 的 ClassWizard 类似,它可以生成对话框和菜单,且能迅速地把 Windows 控制与消息联接起来并定义事件工作的能力。Symantec 的调试器比较出色,它可以检查变化的数据结构,也允许用图像法来分析。

Symantec C/C++ 编译器虽很好,但其不具备优化编译器且不能生成 32 位代码,不过它仍然提供了 Windows 开发的基本调试环境。目前,Symantec C/C++ 已升级到支持 32 位操作的 7.0 版。

4. MetaWare High C/C++ 3.1

MetaWare High C/C++ 3.1 是一种较简单的编译器,它只有一个既与 Pharlap 的 DOS 扩展器兼容又与 Windows 3.x 兼容的产生 32 位代码的编译器,一个 32 位扩展 DOS 或 Windows 应用调试器,一个手工的 UNIX 风格的 DOS 指令工具库。因为 High C/C++ 3.1 只生成使用平面地址模式的 32 位代码,生成一个 Windows 应用程序把资源和保护模式可执行文件与 Windows 扩展器联接起来,这是通过所提供的 MetaWare 的 ADK 的一部分来完成的。High C/C++ 3.1 也可以生成 32 位 DLL 程序,也能够产生优化的 Pentium 代码。

High C/C++ 3.1 是高性能的 32 位程序包,它能最好地配合 CPU 内部操作。目前,High C/C++ 已升级到支持 32 位操作的 3.3 版。

5. Watcom C/C++ 32

Watcom C/C++ 32 编译器的特色是支持更多的操作系统。在这些操作系统中第一个是扩展 DOS。Watcom C/C++ 32 可生成 Windows 3.x 和 Win32s 的代码。

用户可在 DOS 提示符下进行编译,同时,因为编译器输出希望运行在一个 32 位平面地址模式,必须把应用程序和资源联结成一个所需的 Windows 扩展器。尽管 Watcom 软件包也有一些工具,但遗憾的是它没有集成开发环境。它的主要特色是支持多个操作系统平台。目前,Watcom C/C++ 已升级到支持 32 位操作的 10.0 版。

6. Visual Basic 3.0

Visual Basic 3.0(VB)提供了一套全新的图形用户界面编程工具和环境。程序员可以利用这些工具很方便地在屏幕上创建图形用户界面。Visual Basic for Windows 是一种完全支持结构化编程的高级语言。它综合运用了 BASIC 语言及新的可视设计工具。既保留了 Windows 优越的图形工作环境又保留了 BASIC 语言编辑程序的简单性。利用 VB 开发的应用程序可以支持 Windows 的许多高级功能,如多文档界面(MDI)、对象链接与嵌入(OLE)、动态数据交换(DDE)等。总之,VB 适合于非专业程序员在 Windows 下开发应用软件。

7. 其它开发工具

除了以上介绍的几种开发工具外,Microsoft 公司还在 Windows 环境下开发了一些专门的开发工具,如电子表格和商业图形软件 Excel for Windows、数据库语言 FoxPro for Win-

dows 等。Excel for Windows 具有极强的分析性能、报表制作与简报工具,以及丰富易用的统计图表等,它是 Windows 环境下最佳的电子试算表软件。FoxPro for Windows 则是从 MS-DOS 上移植过来的,它是目前 PC 机上速度最快、数据类型最丰富的关系型数据库管理系统,是 XBase 用户和开发人员最有效的升级方案。目前各大计算机软件公司正在不断推出 Windows 环境下的开发工具或将其它操作系统下的优秀软件移植到 Windows 环境下。

2.2 最佳开发语言 Visual C/C++

由于 Microsoft 公司开发了 Windows,自然,一般认为它所开发的 C/C++ 编译器最适合 Windows 环境,随着开发工具的发展,事实也证明确实是如此。综合地说,Microsoft Visual C++ 是在 Windows 环境下完整运行的、紧密集成的开发系统,系统生成代码规模小,运行速度快,调试工具功能齐全,是开发应用软件的强大工具。其主要特点表现在:采用全集成的 Windows 宿主开发工具和对传统的 C/C++ 开发过程采用流行的“可视”用户接口驱动范例。有鉴于此,本书推荐使用 Visual C/C++ 作为 Windows 的开发语言。

2.2.1 Visual C/C++ 的基本构成

Microsoft Visual C/C++ 由以下几部分构成:

1. 可视化工作平台(Visual Workbench——VWB)

Microsoft Visual 工作平台是 Visual C/C++ 开发环境的基础。这个强大的开发环境包括几个集成工具,如编辑器、调试器和图形化浏览器等。Visual 工作平台也可以通过充当中心点来开发更大型的任务。由于采用 Windows 用户界面、集成调试器及后台编译技术(在后台编译的同时,前台可做其它工作),使图 2.1 所示可视化工作平台远胜于其它公司的 C/C++ 开发环境。

在可视化工作平台上可进行编辑(File、Edit 项)、编译(Project 项)、链接(Project 项)、调试(Debug 项)、浏览(Browse 项),还可以启动 AppStudio 资源编辑器(Tools 项)及 AppWizard (Project 项)、ClassWizard(Browse 项)、MFC C++ 开发工具。特别值得一提的是,Visual 工作平台包含了一个完全符合 Windows 规范的文本编辑器。该编辑器功能非常强大,当在 Visual 工作平台下对程序进行编译时,如果发现错误,该编辑器会将出现错误的文本行加亮显示,非常便于调试。另外,该编辑器将注释、C++ 语法构成部分等在显示时分别标以不同的颜色,使人一目了然。

2. 资源编辑器(AppStudio)

如图 2.2 所示的全新的一体化资源编辑器 AppStudio 能在一个集成环境下设计、编辑几乎所有的用户界面对象,包括对话框、菜单、位图、图标、光标、串表、加速键表等。它能单独作为一个资源编辑器为使用 C 及 SDK 来开发 Windows 应用建立资源文件,又能与 AppWizard、ClassWizard 及 Visual 工作平台一起开发基于 MFC 框架的 C++ 应用程序。AppStudio 用作开发 MFC C++ 应用程序时可直接在对话框中加入 Visual Basic 控制并与 C++ 代码相连接。它的最显著特点是可视化(Visualization)。在 AppStudio 下创建资源,无需编写 RC 文