

Visual C++

系列丛书之三

Windows  
Visual  
MFC  
技术与

环境下  
C++  
编程  
实例

李奇 田舒 编著



北京大学出版社

383013

# Windows 环境下的 Visual C++ MFC 编程技术与实例

李 奇 田 舒 编著

北京大学出版社  
北 京

新登字(京)159号

JS195/28

图书在版编目(CIP)数据

Windows 环境下的 Visual C++MFC 编程技术与实例/李奇,田舒编著.  
—北京:北京大学出版社,1995  
ISBN 7-301-02828-8

I. W… I. ①李… ②田… III. C语言-程序设计 IV. TP312C

书 名: Windows 环境下的 Visual C++MFC 编程技术与实例

著作责任者: 李奇 田舒

责任编辑: 沈承凤

标准书号: ISBN 7-301-02828-8/TP·253

出版者: 北京大学出版社

地址: 北京市海淀区中关村北京大学校内 100871

电话: 出版部 2502015 发行部 2559712 编辑部 2502032

排印者: 北京大学印刷厂

发行者: 北京大学出版社

经销者: 新华书店

787×1092毫米 16开本 25.5印张 624千字

1995年5月第一版 1995年5月第一次印刷

印 数: 0000—3,500册

定 价: 49.00元

## 前 言

MS Windows 3+的推出可以说是微机操作系统变革中的一个里程碑。然而,Windows的易学好用,并没有使Windows软件开发的难度减轻。Windows中种类繁多的句柄(Handle)、千变万化的消息(Message)和难以捉摸的事例(Instance),往往把编程人员弄得昏头转向。不仅如此,在Windows应用程序,尤其是有关Windows用户界面的程序的开发过程中,存在大量的重复性劳动,代码规模十分庞大。为了改善这种状况,把编程人员从繁杂的Windows事务中解脱出来,以便让他们把主要精力集中在应用本身的问题上,Microsoft推出了以MFC(Microsoft Foundation Classes)为特色的面向对象的Windows编程环境——Visual C++。

整个MFC类库构成了Windows应用程序的应用框架(Application Framework),同时,Visual C++还提供了在此框架上编程的三位一体的工具:AppWizard、Class Wizard和Studio。用户可以利用AppWizard来自动产生Windows应用程序的框架(即起始代码),也可以利用Class Wizard来管理应用程序中与Windows有关的各种类、消息和处理函数,还可以利用Studio以可视化的方式创建和管理各类资源。借助这些工具,用户只需加入与具体应用有关的代码和资源,大部分与Windows有关的代码和资源都由MFC提供,从而大大提高了应用程序的开发效率。

本书较完整地阐述了MFC应用框架的基本结构,并通过实例循序渐进地介绍了如何运用该类库和辅助工具来编写标准的Windows应用程序,同时还通过剖析一些各具特色的示范程序,介绍了一些常用的编程技巧,最后还列出了一些有用的参考资料。通过这些介绍,特别是对具体实例的研究,相信你会很好地掌握这一新的神奇的编程方法。

值得一提的是,Visual C++ MFC编程环境,反应了CASE这一当代计算机软件科学前沿的某些思想,例如代码自动生成,可视化编程和辅助工具集成等,这对于从事CASE研究的人员也有一定的借鉴意义。

编 者  
1994.12.1

# 目 录

## 第一部分 Windows 编程快速入门

<b>第一章 从程序员角度看 Windows</b> .....	3
1.1 Windows 的由来 .....	3
1.2 窗口解析 .....	4
1.3 Windows 环境下的程序设计要素 .....	6
1.4 遵循 Windows 约定 .....	8
<b>第二章 Windows 编程基础</b> .....	9
2.1 Windows 编程新特点 .....	9
2.2 Hungarian 表示法 .....	10
2.3 WinMain 和窗口过程 .....	11
<b>第三章 从面向对象的观点看 Windows</b> .....	18
3.1 对象和消息 .....	18
3.2 类 .....	20
3.3 对象与数据 .....	22

## 第二部分 MFC Windows 编程起步

<b>第四章 MFC 概述</b> .....	27
4.1 采用 MFC 编程的必要性 .....	28
4.2 MFC 库设计的考虑 .....	30
4.3 MFC 库的主要特点 .....	30
4.4 一切从 CObject 开始 .....	31
<b>第五章 最短的 MFC Windows 程序</b> .....	35
5.1 程序 MINI 清单 .....	35
5.2 关于 AFXWIN.H .....	36
5.3 运行 MINI 程序 .....	36
<b>第六章 MFC Windows 编程示范</b> .....	37
6.1 关于应用程序 Hello .....	37

6.2	如何写 Hello 程序 .....	38
6.3	创建应用对象 .....	39
6.4	在屏幕上显示窗口 .....	42
6.5	与 Windows 通讯 .....	45
6.6	画窗口内容 .....	48
6.7	加入 About 对话框 .....	50
6.8	准备支持文件 .....	51
6.9	创建可执行文件 .....	52
6.10	HELLO 的工作原理 .....	53
<b>第七章</b>	<b>程序清单 .....</b>	<b>56</b>
7.1	Generic 程序清单 .....	56
7.2	Hello 程序清单 .....	60

### 第三部分 MFC Windows 应用框架

<b>第八章</b>	<b>MFC 框架与 Windows 应用程序 .....</b>	<b>67</b>
8.1	MFC Windows 应用框架综述 .....	67
8.2	在应用框架上建造 Windows 应用程序 .....	71
8.3	应用框架如何调用编程者提供的代码 .....	74
8.4	CWinApp 应用类 .....	75
8.5	文档模板 .....	78
8.6	创建你自己的窗口 .....	81
8.7	图形对象(Graphic Objects) .....	86
8.8	如何运用剪接板 .....	88
<b>第九章</b>	<b>消息和命令 .....</b>	<b>90</b>
9.1	应用框架里的消息和命令 .....	90
9.2	应用框架对处理函数的调用 .....	93
9.3	应用框架对消息映射的搜索 .....	95
9.4	消息映射的入口 .....	97
9.5	消息处理函数的声明 .....	98
9.6	如何利用 Class Wizard 来管理命令和消息 .....	99
9.7	用户接口对象的更新 .....	100
9.8	如何在状态条上显示命令信息 .....	101
<b>第十章</b>	<b>框窗、文档和视图 .....</b>	<b>103</b>
10.1	框窗(Frame Windows) .....	103
10.2	文档(Document)和视图(View) .....	107
10.3	打印和打印预观察(Print Preview) .....	115
<b>第十一章</b>	<b>对话框、控制、控制条和上下文相关帮助 .....</b>	<b>117</b>

11.1	对话框	117
11.2	控制(Controls)	126
11.3	控制条(Control Bars)	129
11.4	上下文相关帮助(Context-Sensitive Help)	132

## 第四部分 MFC Wizard 编程

<b>第十二章</b>	<b>创建起始应用程序</b>	<b>139</b>
12.1	创建 RICHE 的起始应用程序	140
12.2	编译起始文件	143
12.3	运行起始应用程序	143
<b>第十三章</b>	<b>创建文档和视图</b>	<b>145</b>
13.1	文档和视图	145
13.2	Riche 的文档:CRichDoc 类	147
13.3	文档数据:CStroke 类	150
13.4	文档管理	152
13.5	数据的连载(Serializing)	155
13.6	Riche 的视图:CRichView 类	158
13.7	视图的重画	159
13.8	在视图里处理 Windows 消息	161
13.9	编译测试 Riche	165
<b>第十四章</b>	<b>构造用户接口对象</b>	<b>167</b>
14.1	编辑 Riche 的菜单	167
14.2	编辑 Riche 的工具条(Toolbar)	172
14.3	Riche 中命令的联结(Binding)	176
14.4	用户接口对象的更新	182
14.5	编译新的 Riche	186
<b>第十五章</b>	<b>加入对话框</b>	<b>188</b>
15.1	对话框的设计	188
15.2	将对话框连结到一个对话类	190
15.3	消息处理函数的实现	200
15.4	对话框的调用	201
15.5	编译新的 Riche	203
<b>第十六章</b>	<b>增强视图功能</b>	<b>204</b>
16.1	多视图的更新	204
16.2	滚动的加入	209
16.3	给 Riche 加上滚动功能	210
16.4	分割窗口的实现	215

16.5	编译新的 Riche .....	219
<b>第十七章</b>	<b>增强打印功能 .....</b>	<b>220</b>
17.1	打印结构.....	220
17.2	增强 Riche 的打印功能 .....	224
17.3	打印预观察(Print Preview)的结构 .....	229
17.4	增强 Riche 的打印预观察功能 .....	230
17.5	编译新的 Riche .....	230
<b>第十八章</b>	<b>加入上下文相关帮助 .....</b>	<b>232</b>
18.1	任务分工.....	232
18.2	利用 AppWizard 实现上下文相关帮助 .....	233
18.3	在 Riche 中加入帮助 .....	238

## 第五部分 常用编程技术

<b>第十九章</b>	<b>定做菜单和控制 .....</b>	<b>249</b>
19.1	属主画(Owner-Drawing)控制/菜单 .....	249
19.2	自绘(Self-Drawing)控制和菜单 .....	249
19.3	自绘控制和菜单示例.....	253
19.4	动态子类化(Dynamic Subclassing)技术 .....	276
19.5	程序清单.....	276
<b>第二十章</b>	<b>定做控制条 .....</b>	<b>277</b>
20.1	实例 CTRLBARS 介绍 .....	277
20.2	多控制条的安排.....	279
20.3	定做的 Tool Bar .....	280
20.4	定做的 Status Bar .....	283
20.5	Dialog Bar .....	284
20.6	浮动工具板(Floating Tool Palette) .....	286
20.7	程序清单.....	288
<b>第二十一章</b>	<b>分割窗口 .....</b>	<b>289</b>
21.1	分割窗口的风格.....	289
21.2	有关术语和对象.....	290
21.3	共享滚动条.....	291
21.4	最小尺寸.....	291
21.5	实际尺寸与理想尺寸.....	291
21.6	保护接口.....	292
21.7	实例 VIEWEX .....	292
21.8	程序清单.....	297
<b>第二十二章</b>	<b>DDX 和 DDV .....</b>	<b>298</b>



22.1	对话数据交换(DDX)的一般形式 .....	298
22.2	对话数据交换(DDX)的工作原理 .....	299
22.3	用户扩充对话数据交换 .....	300
22.4	Class Wizard 对对话数据交换的支持 .....	300
22.5	定做 DDX 的例子 CHKBOOK .....	302
22.6	程序清单 .....	306
<b>第二十三章 MFC 的动态链接技术 .....</b>		<b>307</b>
23.1	一般动态连接库概念与技术 .....	307
23.2	利用 MFC 创建用户 DLL .....	315
23.3	DLL 版本的 MFC .....	320
23.4	程序清单 .....	330
<b>第二十四章 对象链接和嵌入(OLE)技术 .....</b>		<b>331</b>
24.1	OLE 的基本概念 .....	331
24.2	OLE 类 .....	333
24.3	客户应用程序的实现 .....	335
24.4	服务器应用程序的实现 .....	340
24.5	OLE 函数调用的顺序 .....	344
24.6	例子程序 .....	346
<b>第二十五章 VBX 控制 .....</b>		<b>348</b>
25.1	概述 .....	348
25.2	运行时 VBX 支持的初始化 .....	349
25.3	利用 Class Wizard 建立 VBX 控制 .....	350
25.4	加入创建和使用 VBX 控制的代码 .....	352
25.5	VBX 控制的分发 .....	353
25.6	程序清单 .....	353

## 第六部分 一般参考资料

<b>第二十六章 一般用途的类 .....</b>		<b>357</b>
26.1	内存管理 .....	357
26.2	时间和日期 .....	359
26.3	串(String) .....	360
<b>第二十七章 CObject 类 .....</b>		<b>364</b>
27.1	从 CObject 导出一个类 .....	364
27.2	运行时类信息的存取 .....	366
<b>第二十八章 集 .....</b>		<b>368</b>
28.1	如何构造类型安全的集 .....	368
28.2	对集的所有成员的访问 .....	369

<b>第二十九章 文件与连载</b> .....	374
29.1 文件(File) .....	374
29.2 连载(Serialization) .....	375
<b>第三十章 诊断</b> .....	383
30.1 调试特性 .....	383
30.2 内存漏洞的检测 .....	387
30.3 使用 DEBUG_NEW 来帮助调试 .....	391
<b>第三十一章 异常</b> .....	392
31.1 MFC 的异常处理 .....	392
31.2 异常的捕获 .....	393
31.3 检查异常内容 .....	394
31.4 在异常中释放对象 .....	394
31.5 从你自己的函数里引发异常 .....	396
31.6 构造函数里的异常 .....	396
31.7 框变量与异常 .....	397

# 第 一 部 分

## Windows 编程快速入门

第一章 从程序员角度看 Windows

第二章 Windows 编程基础

第三章 从面向对象的观点看 Windows



# 第一章 从程序员角度看 Windows

图形用户接口(GUI)无疑将是未来微机上用户界面的主流。让我们看看 Microsoft Windows 成功之路,就会更坚信这一点。Windows 3.0 软件包在它刚问世之后的前六周就卖出了 500 000 份拷贝,而到第九个月它的销售量已经达到 3 000 000,从而创造了软件包销售史上的新纪录。

Windows 跟 DOS 有很大不同,最重要的不同在于 Windows 是图形用户接口(如图 1-1),它带来了许多新的概念。在 Windows 这里,所有的选择几乎都是以对象(Object)的形式放置在屏幕上,就像一个打开的工具箱一样。这给用户带来了极大的便利,用户不再需要去记住一大堆复杂的技术和关键词,而只须根据任务的需要从手边选取适当的工具。图形用户接口实现了计算机与用户之间空前友好的交流。

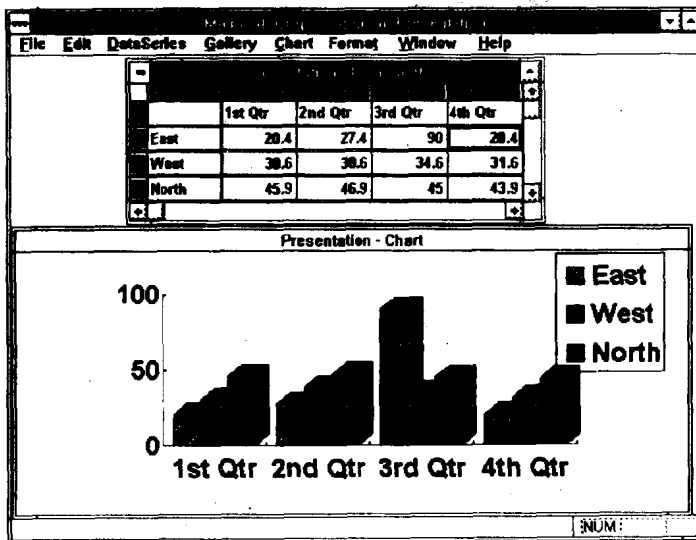


图 1-1 典型的 GUI

## 1.1 Windows 的由来

实际上从 1983 年,也就是 PC 机才问世两年后开始,Microsoft 就开始了 Windows 开发工作。但是它最初的版本 Windows 1.01 直到 1985 年才推出。这一版本是设计运行在当时标准机器,也就是具有 8088 芯片、256K 内存和两套 360K 软盘驱动器的 IBM PC 机上的。它的显示自动安排标题,窗口占据整个屏幕,给人平面感很强。这一版本并没能引起人们多大的注意。

两年之后,它的第二个版本 Windows 2.0 推出了。Windows 第一次能在屏幕上产生重叠

式窗口。然而 Windows 2.0 只能以 80×86 实模式运行,这就意味着它将受到一兆内存的限制。Windows 曾一度被分为 Windows286 和 Windows386,使它能充分利用当时较新的 80386 的新特性。尽管取得了一些新进展,但对于 Windows 2.0 来说有待改进的东西仍很多。

1990 年 5 月,Microsoft 推出了 Windows 3.0。Windows 3.0 给人以全新的感觉,它采用比例字库(Proportional Fonts),使显示更精美。3.0 版本对 DOS 程序也有更好的支持。3.1 版是在 1992 年推出的,它对 3.0 版作了许多改进,尤其是在文件管理方面,很多用户现在都在用 Windows 作为他们 PC 机的主要操作环境。

早先版本的 MS-DOS 运行机制被 Windows 三维一体的管理机制:程序管理器(Program Manager)(见图 1-2)、任务表(Task List)(见图 1-3)和文件管理器(File Manager)(见图 1-4)所取代。从编程的观点来看,Windows 3+最重要的特点就是它们能支持扩展内存,最大可达 16 兆字节 RAM。与此同时,在增强 386 模式下,Windows 利用 803/4/586 内部的虚拟存储功能(也就是将内存的部分暂时存放在磁盘上),使编程者能访问到比实际物理内存大 4 倍的地址空间。譬如在一台内存 16 兆字节的机器上,Windows 可以为编程者提供 64 兆字节的存储空间。应该说 Windows 3+最终达到了 Windows 设计的目标。

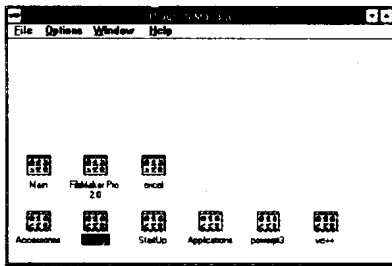


图 1-2 程序管理器

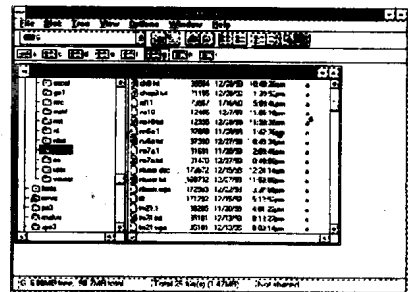


图 1-3 文件管理器

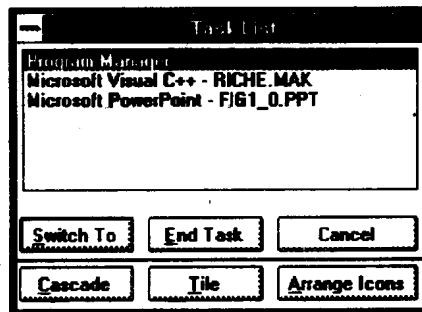


图 1-4 任务表

## 1.2 窗口解析

一个典型的 Windows 3+窗口如图 1-5 所示。在我们继续之前,有必要熟悉一下窗口的各个组成部分以及它们的名字。在编程之前了解用户希望从你的 Windows 应用程序中得到什么东西是很重要的。

在图的左上角是系统菜单框(System Menu Box),当被选中时,它会显示一菜单。典型的系

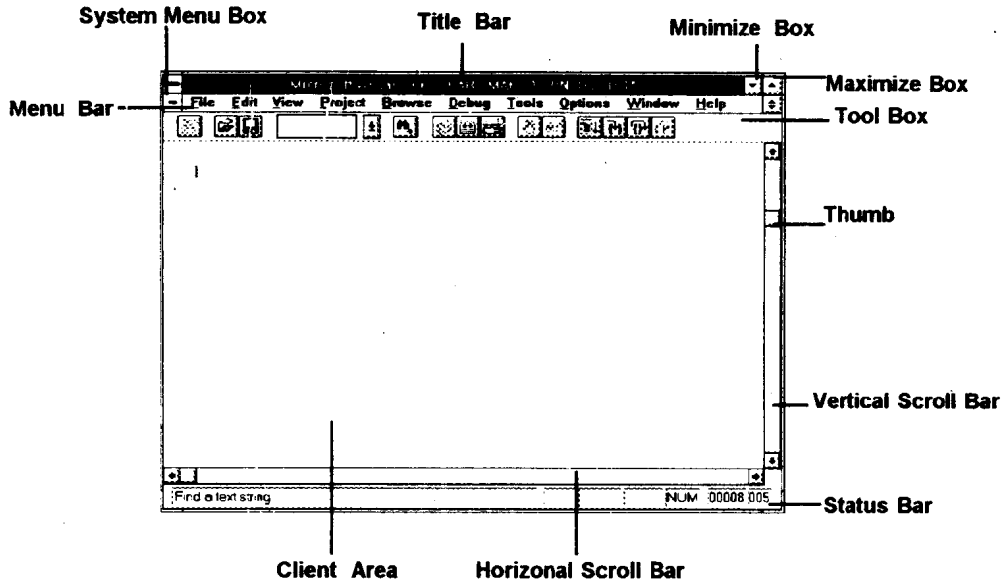


图 1-5 Windows 3+ 窗口

系统菜单提供的功能选择有移动窗口、关闭窗口、最大化窗口和最小化窗口。在顶部中间是标题条(Title Bar),一般用来标明某个应用程序。

在标题条的右边是最大化框(Maximize Box)和最小化框(Minimize Box),用户可以通过点击它们分别将窗口缩小成一个图符(Icon)(称为应用程序的图符状态),或扩展到整屏。在标题条之下通常是菜单条(Menu Bar),用来显示当前应用程序所提供的功能选择。如图 1-6 所示。

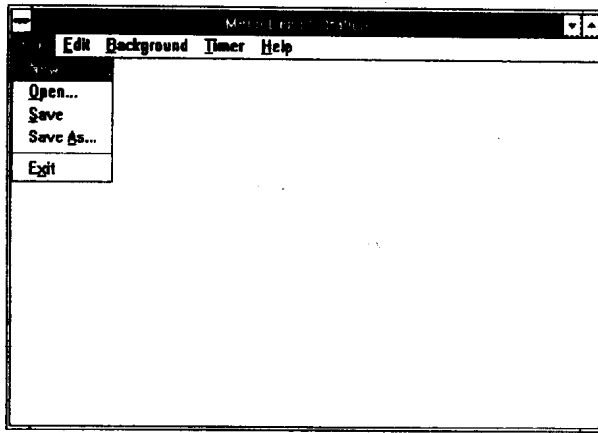


图 1-6 具有菜单的 Windows 3+ 窗口

图中 Exit 项通常用来结束应用程序。因此,如果你的应用程序支持文件处理的话,应当在 File 菜单最底部包含 Exit 项。实际上,即使不用到文件处理,Windows 应用程序也经常有一个 File 菜单并提供 Exit 项。

在菜单条下边是工具条(Tool Bar),它以非常形象的按钮方式将应用程序的功能选择列于其上,用户通过点击中相应的按钮就能选中相应的操作,工具条上列出的选项一般说来在菜单里也都提供,之所以另外做成工具条,是因为它们使用的频率较高,从工具条上选中比从菜单条上选中更直观、更快捷。窗口最底部是状态条(Status Bar),用来显示各种状态信息,譬如正

在进行的操作、系统内存、磁盘、键盘等等状态信息。工具条和状态条通常是可拆卸的。

菜单条下是客户区(Client Area)。这是我们可写可画的区域,也就是说,在窗口的这一部分,我们将可放置按钮(Buttons),列表框(List Boxes),正文框(Text Boxes)以及其它东西。在Windows里这些可见的对象被称为控制(Controls)。

在客户区的右边是垂直滚动条(Vertical Scroll Bar),在显示文档(Document)的窗口里它通常要被用到。如果文档太大以至窗口一次容纳不下,你可以通过滚动条在整个文档里来回滚动,看到你想看的部分。客户区的底部是另一种滚动条,水平滚动条(Horizontal Scroll Bar),让文档在客户区里水平滚动。

窗口里客户区之外所有部分被称为非客户区域,甚至连边框也属于非客户区域。Windows将负责维护窗口的非客户区域,编程者负责客户区。

### 1.3 Windows 环境下的程序设计要素

当我们开始 Windows 程序设计的时候,首先面对的是一个固定的空窗口(如图 1-7)。

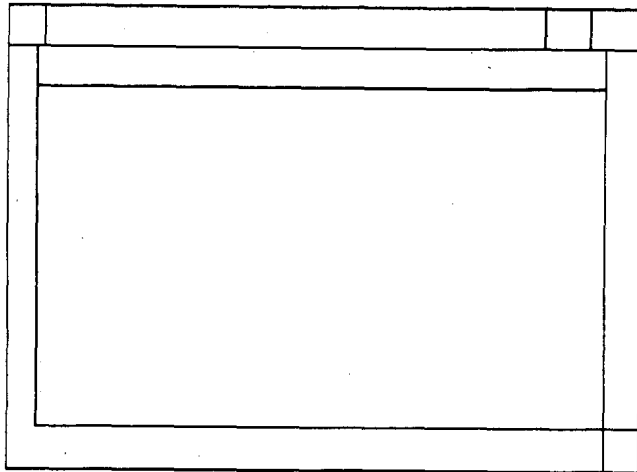


图 1-7 一个空窗口

除此之外,你可能想在你的主窗口里放上一些按钮(Buttons)来指示用户可选操作,但最好不要这么做,因为通常象按钮、列表框和组合框(Combo Boxes)都用在对话框或子窗口里,而不是主窗口里。主窗口里提供的选择通常是菜单形式(如图 1-8)。

当用户将鼠标(Mouse Cursor)移到菜单名上并按下鼠标键时,这一菜单被打开(如图 1-9)。

菜单项后面的点表示,如果选中他们将会弹出一对话框。你的应用程序里可以随意弹出任意多的对话框(如图 1-10),对于用户来说,对话框可能是最有用的东西。有的编程者得出一条经验:除了正文框(Text Box)和菜单(Menu),任何其它的控制(Control)都应当在对话框里出现。

另外,我们还将要搞清楚如何去变灰(也就是使它变为不活动的)菜单项,这是因为在某些情况下这些菜单项不再能提供合法的选择,譬如,在没有任何文本存在时,选择“Save File As ...”显然是不合法的。但是,在需要变灰的菜单项很多的情况下,你最好不用这种方法,而索性



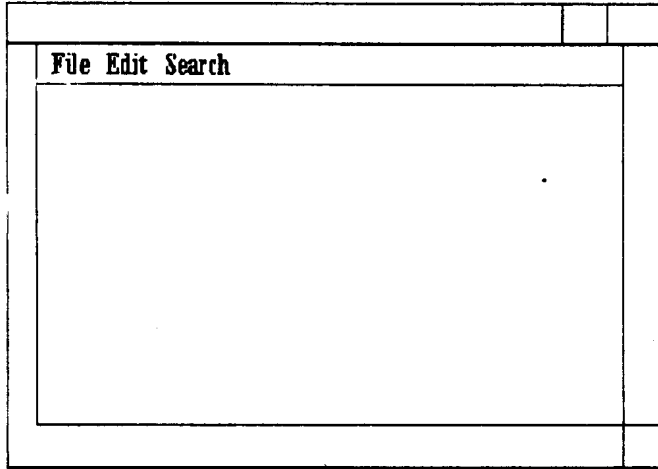


图 1-8 一个具有菜单的窗口

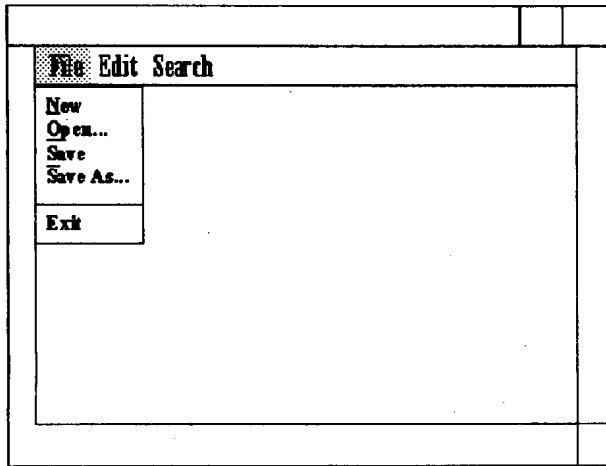


图 1-9 一个具有打开的菜单的窗口

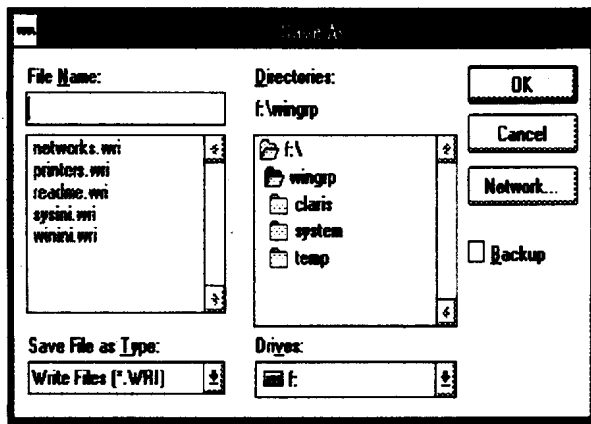


图 1-10 一个典型的对话框

将这些项从菜单中移去。

与主窗口相反,对话框可以拥有任意多的控制(注意:对话框的边框是单线框,与主窗口的