

Windows 会计信息系统教程



王永生 主编

21 世纪会计电算化系列教材

Windows 会计信息系统教程

王永生 主编

黄平生 叶小玲 副主编

科学出版社

北 京

内 容 简 介

本书是“21 世纪会计电算化系列教材”之三, 本书以从会计业务要求的角度提出问题, 从可视化编程的角度解决问题的原则, 全面系统地讨论了多媒体会计信息系统的基本概念、运行维护、当今会计软件开发与发展前景。内容包括 Windows 环境与机制、面向对象程序设计、Visual FoxPro 可视化编程技术、多媒体与网络技术、应用程序开发、会计信息系统分析与开发方法、账务子系统设计与开发、会计决策支持系统。

本书章节编排顺序符合人的逻辑思维方式, 内容由浅入深, 循序渐进, 易于理解记忆, 同时穿插丰富的开发技术实例及完整的源程序, 并配有适量的练习题与上机操作, 便于学习、仿效、操练和掌握书中的新方法、新技术、新知识。通过本书学习, 读者可以具备开发多媒体会计信息系统的能力。

本书可作为大专院校财会专业会计信息系统(或会计信息系统开发)课程的教材, 亦可作为在职会计人员岗位培训用书, 还可供计算机信息系统开发人员或有关软件技术人员参考。

图书在版编目(CIP)数据

Windows 会计信息系统教程/王永生主编. —北京: 科学出版社, 2003
(21 世纪会计电算化系列教材)

ISBN 7-03-011607-0

I. W... II. 王... III. 会计—管理信息系统—教材 IV. F232

中国版本图书馆 CIP 数据核字(2003)第 044647 号

策划编辑: 陈晓萍 / 责任编辑: 丁 波

责任印制: 吕春珉 / 封面设计: 王 浩

科学出版社 出版

北京东黄城根北街16号

邮政编码: 100717

<http://www.sciencep.com>

双青印刷厂 印刷

科学出版社总发行 各地新华书店经销

*

2003 年 6 月第 一 版 开本: 787×1092 1/16

2003 年 6 月第一次印刷 印张: 18 1/2

印数: 1—4 000 字数: 422 000

定价: 26.00 元

(如有印装质量问题, 我社负责调换<路通>)



前言

教育具有先进性与超前性,这是共识。但是由于科学技术飞速发展,目前教学内容的陈旧和滞后现象还会存在。要解决这个问题,现代课程设置就应当是开放的、灵活的,不应是依据预先规定好的教学目标而一成不变,必须不断更新,开创新的课程体系。

为满足新世纪对人才知识、能力、素质的要求,我们将财会本科学生培养成既懂会计又懂计算机的高素质人才,以适应社会科学技术发展对会计复合型人才的需要。目前,财会专业会计电算化的教学水平与我国经济技术发展的实际需要、新世纪对人才的要求还有很大差距,课程设置起步较晚,所授课程不尽相同,课程内容也相差较大。有的专业把会计电算化课程放在一起上,内容较多,显得杂乱,不便于学生系统掌握;有的专业虽然分开上,但是没有系列教材,还处在单本状态,不能很好地衔接,影响整体教学效果。将会计电算化课程系列化,融会贯通,前后课程互相衔接,形成会计电算化系列课程体系,这是搞好会计电算化教学改革、提高教学水平、进一步发展的必然之路。

建设会计电算化系列课程体系首先要创建会计电算化系列课程教材。要提高教学质量,关键在教师和教材,教材是教师授课取材之源,是学生求知复习之本。不问教学内容,不提供教学范本,空喊提高教学质量,肯定于事无补。显然,丢开教材,欲求提高教学质量,不啻缘木求鱼。会计电算化教学内容要提高一个层次,使学生学到一定的深度和广度,毕业后在实际工作中才能应付自如,表现出很强的后劲。将来的开发、维护会计软件的人员应该主要是学会计专业的,而不应该是学计算机专业的,会计专业人员要“自己救自己”,这与当前会计电算化教学存在很大的差距。

会计电算化软件已进入 Windows 多媒体环境,从而导致会计电算化教材工作迈进了一个新阶段,并面临着开创性的任务。

基于上述种种情况,要求我们密切结合课程设置,重视学生的认知规律,吸收教学研究的成果,抓好会计电算化的教材建设,内容要有系统性、先进性、新颖性、实用性,整体处理要体现自己的特色。

为此,作者经过精心策划,结合多年教学、研究、应用的经验,编著了面向新世纪的、力争国内一流水平的“21 世纪会计电算化系列教材”中的三本书——《Windows 会计电算化基础》、《Windows 电算化会计实务》和《Windows 会计信息系统教程》,以满足 Windows 多媒体环境会计电算化教学与研究发展的迫切需要。同时,这也满足会计电算化初、高级教学内容的要求。初级教学内容侧重于会计软件的操作使用,而高级教学内容则还应包括会计软件的运行管理与维护开发。其中,《Windows 会计电算化基础》和《Windows 电算化会计实务》组成初级教学内容,全套系列教材构成高级教学内容。

本书是“21 世纪会计电算化系列教材”之三,它基于 Windows 2000 操作系统,使用 Visual FoxPro 编程工具开发会计信息系统。书中既讲述会计信息系统的整体框架、各

子系统的工作任务及工作流程等，又实打实地告诉读者如何从会计业务出发、利用计算机技术开发一个完全能投入使用的多媒体会计信息系统。即基本上遵循从会计业务要求的角度提出问题，从可视化编程的角度解决问题的原则，从会计信息系统的分析与设计入手，开发系统多媒体主界面、凭证输入、凭证修改、凭证删除、凭证查询、科目汇总、记账、结账、查账等软件。学完本书，相信读者对于设计一个多媒体会计信息系统会有很清楚的认识，软件开发能力会有显著的提高，进而成为“身手不凡”的会计电算化人才。

全书共分 10 章：第 1 章简介 Windows 环境与机制，第 2 章讲解程序设计，第 3 章论述面向对象程序设计，第 4 章讲述控件使用，第 5 章介绍多媒体与网络技术，第 6 章概述应用程序开发，第 7 章阐述会计信息系统及其发展，第 8 章讨论会计信息系统的开发方法，第 9 章介绍账务子系统设计与开发，第 10 章叙述会计决策支持系统。书中各章均配有适量练习与上机题。可安排 60 个学时，最好是边讲课，边上机练习。

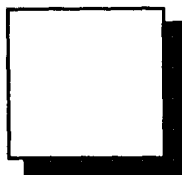
本书由王永生主编，黄平生、叶小玲为副主编，参加编写工作的还有艾志刚、王燕、王正花、吴晓琳、郭昊等。

本书献给从事会计电算化教学事业的师生以及所有从事这一领域工作和研究的同行。

书中难免有错，敬请读者指正。

作 者

2003 年 3 月



目 录

1	Windows 环境与机制	1
1.1	Windows 环境	1
1.2	Windows 机制	5
1.3	Windows 开发工具	9
	练习题	11
2	程序设计简介	12
2.1	程序设计的优点	12
2.2	VF 的编程机制	13
2.3	程序设计的基本概念	15
	练习题	20
3	面向对象程序设计	21
3.1	面向对象程序设计的基本概念	21
3.1.1	对象	21
3.1.2	类	22
3.1.3	事件	23
3.1.4	控件	24
3.1.5	容器	24
3.1.6	对象的属性	25
3.1.7	对象的方法	25
3.1.8	对象的事件	25
3.1.9	对象引用	27
3.1.10	设置属性	29
3.1.11	响应事件	30
3.1.12	调用方法	32
3.2	画各种控件	32
3.2.1	画控件	32
3.2.2	各种控件	36
3.3	可视化编程	47
3.3.1	布局、设置对象属性	47
3.3.2	控制逻辑与何地代码	51
3.3.3	编写代码	54
3.3.4	编程示例	54

3.4 使用过程和自定义函数	58
练习题	61
4 控件的使用	62
4.1 深入了解控件和数据	62
4.1.1 控件和数据的关系	62
4.1.2 控制对数据的访问	63
4.2 根据任务使用控件	69
4.2.1 使用命令控件	69
4.2.2 使用选项控件	70
4.2.3 使用输入控件	76
4.2.4 使用微调控件	79
4.2.5 使用显示控件	80
4.2.6 使用时间控件	83
4.3 使控件易于使用	84
4.3.1 设置访问键	84
4.3.2 设置控件的 Tab 键次序	84
4.3.3 设置工具提示文本	85
4.3.4 更改鼠标指针显示	85
4.3.5 启用和废止控件	85
4.3.6 允许用户拖放	86
4.4 创建类或控件	86
4.4.1 创建控件	86
4.4.2 修改控件	102
4.4.3 复制、删除或查看类	103
4.4.4 以编程方式定义类	104
4.5 建立任务控件	105
4.5.1 决定什么情况下创建控件	105
4.5.2 控件与任务匹配的实例	106
练习题	109
5 多媒体与网络技术	111
5.1 多媒体编程技术	111
5.1.1 媒体控制接口	111
5.1.2 Windows 应用程序编程接口	115
5.1.3 VF 调用动态链接库函数	115
5.1.4 调用多媒体函数 MciSendString	117
5.1.5 VF 多媒体编程	119
5.2 自制声音图标按钮控件技术	122
5.2.1 定义声音图标按钮控件类	123
5.2.2 定制声音图标按钮控件	127

5.2.3 使用声音图标按钮控件	129
5.3 开发多用户应用程序	130
5.3.1 开发多用户应用程序中遇到的术语	130
5.3.2 多用户应用程序开发中的主要问题	131
5.3.3 编写多用户应用程序	132
练习题	134
6 应用程序开发	135
6.1 应用程序的建立	135
6.1.1 解决问题分析	135
6.1.2 创建数据库	136
6.1.3 可视化编程	136
6.1.4 程序测试调试	137
6.2 应用程序连编	138
6.2.1 应用程序的构成	138
6.2.2 使用项目管理器	141
6.2.3 连编应用程序	143
6.2.4 运行应用程序	145
6.3 应用程序发布	146
6.3.1 准备发布程序	146
6.3.2 建立发布树	148
6.3.3 创建发布磁盘	148
练习题	154
7 会计信息系统概述	155
7.1 会计信息系统的概念	155
7.1.1 会计信息	155
7.1.2 会计信息系统	159
7.1.3 计算机会计信息系统	163
7.2 会计信息系统的核算制度	166
7.2.1 记账方法规范化	166
7.2.2 账户、凭证、账簿和报表的设置	169
7.2.3 账务处理基本模式	176
7.3 会计信息系统的发展前景	183
7.3.1 会计信息处理技术的发展	183
7.3.2 会计信息系统的发展前景	186
练习题	189
8 会计信息系统的开发方法	190
8.1 系统的开发方法	190
8.1.1 原型法	190
8.1.2 面向对象法	192

8.1.3	生命周期法	195
8.2	会计信息系统的总体分析	197
8.2.1	系统目标与规模确定	197
8.2.2	可行性分析	197
8.2.3	业务调查	198
8.2.4	系统的子系统划分	200
8.2.5	数据流图	202
8.2.6	数据词典	208
8.2.7	系统分析报告	214
8.3	会计信息系统的总体设计	215
8.3.1	会计信息系统设计原则	215
8.3.2	系统网络设计	217
8.3.3	系统数据库设计	222
8.3.4	代码设计	224
8.3.5	输入输出设计	225
8.3.6	系统设计说明书	226
8.4	会计信息系统的实施	227
8.4.1	可视化编程	227
8.4.2	系统实施	229
8.5	会计信息系统维护	231
8.5.1	系统评审与评价	231
8.5.2	系统使用	232
8.5.3	系统维护	232
	练习题	233
9	账务子系统的设计与开发	234
9.1	会计科目体系维护功能的设计与开发	234
9.1.1	为什么要对科目编码	234
9.1.2	科目编码方式与编码结构	235
9.1.3	科目代码结构灵活性与助记码	236
9.1.4	校验码设置	236
9.1.5	会计科目体系维护功能设计	237
9.1.6	会计科目体系维护功能的可视化编程	238
9.2	凭证处理子系统总体设计	256
9.2.1	凭证子系统功能分析	256
9.2.2	凭证子系统数据库设计	258
9.3	凭证输入修改模块的设计与开发	261
9.3.1	凭证输入格式设计	261
9.3.2	凭证输入控制设计	263
9.3.3	凭证输入修改模块的功能设计	265

9.4 凭证审核模块的设计与开发	268
9.5 凭证汇总模块的设计与开发	270
9.5.1 凭证汇总功能分析	270
9.5.2 凭证汇总模块功能设计	270
9.6 凭证查询模块的设计与开发	271
9.6.1 凭证查询功能分析	271
9.6.2 凭证查询模块功能设计	271
练习题	272
10 会计决策支持系统	274
10.1 会计决策支持系统的产生	274
10.1.1 现代会计职能要参与决策	274
10.1.2 会计决策支持系统开发势在必行	275
10.1.3 会计电算化发展的高级阶段	276
10.2 决策支持系统组成	277
10.2.1 模型库	277
10.2.2 数据库	278
10.2.3 方法库	278
10.3 会计决策的内容和模型	279
10.3.1 会计决策的内容	279
10.3.2 会计决策模型	280
10.3.3 常用的方法	280
10.4 会计决策支持系统的开发方法	282
10.5 会计决策支持系统举例	282
练习题	284

1

Windows 环境与机制

在 Windows 操作系统下开发会计软件，首先要了解 Windows 的编程环境和机制，然后要选取合适的软件开发工具。本章对这些内容进行较深入一点的讨论，为进一步学习会计信息系统开发方法与编程，奠定良好的基础。

1.1 Windows 环境

Windows 环境是指 Windows 操作系统提供的资源、开发 Windows 应用程序所用的软件工具，以及 Windows 应用程序提供给用户的交互方式，比如菜单、工具上有什么命令或者鼠标经过一个窗口时所具有的形状等。

在 Windows 中，资源不仅包括系统的硬件资源，如显卡、打印机、声卡、视卡等，还包括另一类重要的资源，即程序资源。Windows 中的窗口、菜单、图标、光标、位图、对话框、字符串等都可以以程序资源的形式被应用程序使用。最终，程序资源以二进制的形式存储在 .EXE 或 .DLL(动态链接库)文件中。它们通常驻留在磁盘上，当需要时才调入内存，Windows 内存不足时还可以废弃一些资源以释放内存。资源概念及机制的引入，实际上提高了内存的利用率，方便了应用程序的开发。另外，由于资源和源代码相互分离，故容易将应用程序本地化。如将 Visual FoxPro 汉化，只需将所用的资源翻译为汉字，再将其与其他代码重新编译、连接即可，而不必逐行地查找、修改代码。

有人说 Windows 犹如一株人参，如图 1.1 所示。

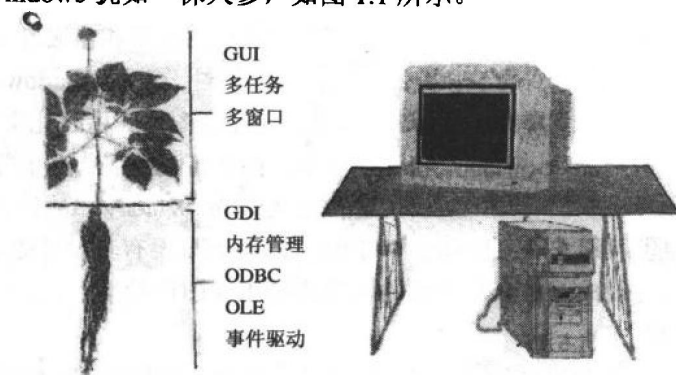


图 1.1 Windows 像一株人参

对用户来讲，它呈现的只是地面之上的漂亮花朵。即从用户的角度，Windows 提供了一致的图形化界面 (GUI)以及多任务、多窗口的环境，使用户只需简单地对图标、对

话框、菜单、按钮等进行选择和操作即可完成所需功能，并能在各个应用程序之间快速切换。但从程序员的角度来讲，为达到上述目的，Windows 提供了与 DOS 截然不同的内存管理方式、事件驱动的编程模式、多任务且同时可运行多道应用程序的环境，使得程序员可以开发出功能齐备、使用方便、界面新颖且美观的应用程序，这对 Windows 本身来说也是一种增值。Windows 程序员最能懂得“人参”地面之下最有价值的东西，以及怎样充分发挥它的功能。由于 Windows 的种种优点，开发符合 Windows 标准的应用程序越来越受到人们的重视。Windows 所定义的图形用户界面以及及设备无关的图形设备接口(GDI)已成为 PC 机事实上的操作系统标准。

1. 图形用户界面(GUI)

Windows 的图形用户界面(GUI, Graphic User Interface)提供了可视化的操作界面。DOS 中涉及的界面元素在 Windows 中被归结为窗口、图标、对话框和菜单，系统环境及资源的管理被大大简化，输入和输出也变得简单、高效。屏幕从 DOS 中的键入命令或文本信息的字符等方式中解脱出来，变为向用户提供了直接、丰富、自然的交互手段。GUI 的作用不只是从 DOS 的字符环境走入了图形环境。作为 PC 机操作系统事实上的标准，GUI 带来了用户界面的一致性。这种一致性使得用户在不同应用程序之中获得的经验可以相互移植，用户不再需要浪费时间去学习不同软件的操作，而是可以举一反三，大大提高效率。从程序员的角度来讲，GUI 的一致性是通过 Windows 来完成的。Windows 为窗口、对话框、菜单等预设了模板，程序员所做的只是填充模板，然后由 Windows 的内部例程实际构造窗口、对话框、菜单等。窗口、图标、对话框、菜单的一些行为，比如打开、关闭、最大化、最小化，也是由 Windows 统一管理的。

图形用户界面可以说是 Windows 环境展现在用户和程序员面前的窗口。当今 Windows 图形用户界面支持图形图象，是多媒体软件开发的理想界面。

2. 图形设备接口(GDI)

在 Windows 中，可以存取丰富的与设备无关的图形操作集。也就是说，应用程序能方便地画直线、矩形、圆和其他复杂的图形。由于 Windows 提供了设备无关性，所以可使用同一函数在打印机或高分辨率显示器上画一个圆。为 Windows 编写的应用程序不直接访问屏幕和打印机等图形显示硬件设备，而是把输出发送到相应硬件设备的设备描述表中。这意味着，应用程序不必关心硬件设备的技术特性。为此 Windows 提供了一种图形设备接口(GDI, Graphic Device Interface)语言，使显示图形和格式化文本变得很容易。GDI 将图形输出发送到硬件设备的设备描述表中，由设备描述表指示的设备驱动程序将图形输出转化为硬件设备的输出，所有的外部设备都靠 Windows 的设备驱动程序来管理，程序员编写的代码可在任意视频卡和打印机上运行(只要有其设备驱动程序)。这样，程序员无需为所有可能的设备开发不同的程序来使用其图形功能。

3. 多任务处理

在传统的 DOS 方式下一次只能运行一个应用程序，该程序独占系统的存储器 and 输入输出设备，要运行其他应用程序必须退出该程序。而在 Windows 中则能够在同一时间内处理多个任务，比如同时打开多个应用程序，每个应用程序占用一个窗口，可以在窗口之间快速切换或传递信息。所有应用程序共享系统资源，Windows 按顺序给每个应用程序分配一段时间来访问 CPU。由于 CPU 切换速度极快，使我们感觉多个任务在同时

进行。

虚拟机(Virtual Machine)是 Windows 实现多任务的核心和关键。它实际上是 Windows 在内存中创建的逻辑微机,用于处理各个应用程序。每个虚拟机都相当于一台独立、完整的微机。虚拟机管理器(VMM, Virtual Machine Manager)管理虚拟机的建立及资源分配,它负责为每台虚拟机分配适当的资源,使每台虚拟机都有自己的内存空间,某些情况下还具有自己的设备驱动程序和环境变量。在 Windows 3.X 中,所有应用程序共享一个虚拟机,即在同一个虚拟机中运行。这些应用程序以协同的方式,轮流使用 CPU 一段时间,然后释放。另一个再使用,再释放,以次类推,这称为协同式多任务。协同式多任务的缺点是一旦某个应用程序占用 CPU 而不释放,则多任务将被迫中止,直到该程序释放 CPU 为止。例如,在 Word 保存很长的文档时,鼠标变为沙漏状,表明不能进行其他操作。

在 Windows 95 以后版本中,每一个应用程序都在自己的虚拟机中运行。这就意味着,应用程序的执行不必等待正在运行的应用程序释放 CPU。如上例,如果不想在 Word 存盘期间等待,而是“抢先”运行 Excel,Windows 将挂起虚拟机,使之后台执行,同时把系统控制权交与 Excel 的虚拟机。Word 中箭头和沙漏的组合图标表明 Word 正在执行,但可以不等等待而执行其他任务,这称为抢先式多任务。另外,Windows 95 以后版本中各个应用程序的虚拟机彼此分离、互不干扰,一旦某个应用程序发生意外死锁或中止,也不会像 Windows 3.X 那样造成系统崩溃。

4. 信息交换

Windows 为其应用程序之间的信息交换提供了四种标准机制:剪贴板、动态链接库(DLL)、动态数据交换(DDE)和对象的链接与嵌入(OLE)。

(1) 剪贴板是一种静态数据交换技术,可以把一个应用程序中的信息剪裁或复制下来,然后粘贴到其他应用程序的适当位置。

(2) 动态链接库(DLL, Dynamic Link Library)包含了若干程序的可执行模块,Windows 应用程序可以调用这些函数来完成有用的任务。Windows 利用 DLL 来建立 Windows 应用程序,可以利用 Windows 函数和资源。DLL 允许多个应用程序共享例程的单个副本,比如几个程序均使用同一个 Windows 函数,内存中将只存在该函数源代码的一个副本。DLL 还可以用来共享数据和硬件等资源,例如 Windows 的字体实际上是通过 DLL 共享的文本数据。

(3) 动态数据交换(DDE, Dynamic Data Exchange)是在多个应用程序之间,以客户/服务器方式建立起一条数据链路。它是一种进程间通信的方式,通过 DDE,可以将完全不同的应用程序中的数据链接起来,实现二者的通信和数据交换。服务器方的原始数据变化时,可自动更新链接到用户方的数据,实现了不同应用程序之间数据的动态更新。

(4) 对象的链接与嵌入(OLE, Object Linking and Embedding)是 DDE 技术的改进。这里的对象是指由 Windows 应用程序生成的任何信息。例如,Excel 生成的报表或者是画笔生成的图画。产生对象的文档称作源文档,放入对象的文档称作目标文档。嵌入对象就是将一个文档(源文档)中的信息插入到另一应用程序生成的文档(目标文档)中。嵌入的对象与传递此对象的源文档不再有关,对嵌入的对象所做的更改不反应到源文档中。进行对象链接时,并不对信息进行复制,而是对包含要链接对象的文档(源文档)生成一个

引用或链接。编辑链接对象，实际上就是对源文档中的信息进行操作。Visual FoxPro 中包含了工业标准 OLE Automation 技术，借助 OLE 自动化，可以通过编程方式控制其他应用程序，从而扩展应用程序的功能。例如，可以指示 Excel 完成一些统计运算，指示 MS-Graph 画出统计结果的图表，然后，将图表存储到 Visual FoxPro 表的通用字段中，而所有这些工作都可用 Visual FoxPro 代码来实现。

5. 开放数据库连接(ODBC)

目前众多厂商纷纷推出形形色色的关系型数据库管理系统(RDBMS)，如 FoxPro、Oracle、Sybase、Informix 等。在它们忙着争夺市场时，却把不同数据库之间互相访问的棘手问题留给了别人。虽然可以通过 DBMS 提供的 API 实现不同数据库的互联，但程序员需要熟悉多种 DBMS 及其 API，一旦采用另外的 DBMS，还要重新进行编码，这给开发和维护工作带来了很大困难，数据库互联的开放化、标准化势在必行。

1991 年 11 月，微软公司发布了开放数据库连接(ODBC, Open Database Connectivity)，次年推出可用版本。ODBC 采取了数据库驱动的思想，这大概是受了 Windows 中打印驱动思想的启发。在 Windows 中，用户安装不同的打印机驱动程序，使用同一条打印语句或操作，就可以很方便地实现打印输出，而不必了解其内部的机理。有了 ODBC 后，用户安装不同的 DBMS 驱动程序就可使用同样的 SQL 语句在不同 DBMS 上实现同样的操作。

开发厂商根据 ODBC 标准开发底层的驱动程序，用户无需了解它们。ODBC 为用户提供了简单、标准、透明的数据库连接的公共接口，并逐步成为 Windows 及 Macintosh 平台上的标准接口。

6. 内存管理

Windows 突破了 640KB 基本内存限制，能够直接访问 16MB 以上的存储空间，并提供了虚拟内存管理的能力。借助于 Windows 高效的内存管理机制，运行于 Windows 之中的应用程序在任何时候都可以将超过实际内存大小的代码提交给系统运行。Windows 在增强模式下运行时，可以通过“交换”来优化内存的使用。交换能将信息暂时从内存中转移到硬盘上，使内存可供其他程序使用。通过 DLL，多个应用程序可以共享一个程序代码的同一副本，节省了内存。

7. 真型(TrueType)字体技术

真型字体技术即 TrueType 字体技术。计算机系统输出一般有两个地方：屏幕及打印机，相应有两种字体：显示字体及打印字体。在 DOS 中，显示字体与打印字体并不相同。使用过 WPS 的读者可能深有体会，要想在打印机上输出仿宋体，必须在显示字体前加上仿宋体的控制码。Windows 中提供的 TrueType 字体解决了这个问题。TrueType 字体存储字型的轮廓及“线索”。所谓的线索是一些算法，用来改善字体缩放后的外观。TrueType 字体可以被任意放大或旋转，在任何尺寸上，TrueType 字体都有着良好的外观。更重要的是，TrueType 字体在屏幕上显示的与在打印机上打印出来的效果完全相同，真正实现了所见即所得(WYSIWYG, What You See Is What You Get)。

8. 多媒体技术

Windows 提供了多媒体技术的功能，能够支持十多种多媒体设备。用户可以利用声卡、麦克风在 Sound Recorder 应用程序中录制或收听声音；还可以利用 Media Player 操

作各类多媒体文件,运行 CD-ROM 驱动器、MIDI、CD 唱机等多媒体设备。在 Visual FoxPro 中,可以使用 General 字段或 OLE 存储和操作多媒体数据,并且通过 MCI(媒体控制接口)调用 DLL 中 Windows API 函数 mciSendString()编写多媒体应用程序。

9. 网络支持

Windows 是在 PC 机上发展起来的。20 世纪 80~90 年代,随着微机技术的发展,局域网技术得到了迅猛的发展,并成为当今信息界的热点。Windows 也及时向这个领域扩展,Windows 3.X 已经提供了很多功能,使它很容易与网络资源协同工作。从 Windows 95 开始,它在很大程度上是一个网络操作系统,特别是到了 Windows 2000,它已经是一个网络操作系统,既支持工作组级别的组网方式,也支持以客户/服务器方式访问其他网络,这使 Visual FoxPro 可以支持客户/服务器网络解决方案。

1.2 Windows 机制

Windows 机制主要是指事件驱动程序运行的机制。Windows 提供完善的事件处理机制,通过它用户能十分方便地实现对操作的响应。在事件模型中,任务的执行是对事件的响应,应用程序始终在运行中等待事件的发生,然后执行相关的操作。Windows 始终控制整个的事件循环,实现多任务的并行操作、多程序的同时启动运行。

在介绍 Windows 事件驱动机制之前,首先介绍 Windows 中控件及对象的概念。

1. 控件及对象的概念

在 Windows 环境下,图形用户界面提供了用户与应用程序之间的交互,它不仅用作输出设备,还可以用作输入设备,就像真实设备的控制界面一样。这种显示在应用程序的图形用户界面中,可供用户操纵,从而控制应用程序的图形界面元素称为控件(Control)。命令按钮、复选框、列表框等都是控件的例子,它们往往用来接受用户输入或向应用程序传达用户的指令。

Windows 环境下所指的对象(Object)比控件的含义更广泛,对象是对程序员而言的,控件是对应用程序的用户而言的。对象是指程序员在程序设计过程中可以访问的元素,它包括控件所代表的图形对象,还包括窗口、屏幕、打印机等环境对象。一个命令按钮对用户来讲,是向应用程序发命令的控件;而对程序员来讲,是可视化编程中可重用的命令按钮对象。

窗口是 Windows 环境中最重要的元素。从用户的角度讲,窗口是一块工作区,应用程序在其中完成输入输出操作。从程序员的角度讲,窗口是应用程序接收和处理信息的对象,也是一个虚拟的屏幕。Windows 中的窗口有两类:应用程序窗口和包含在应用程序中的窗口。应用程序窗口包含一个正在运行的应用程序,应用程序窗口中的窗口则代表该应用程序的不同线程。从表面上看,应用程序窗口包含应用程序名、相关文档,以及应用程序的菜单栏。应用程序窗口中的窗口则没有菜单栏。应用程序窗口中的窗口虽说是由应用程序创建的,但该窗口的管理实际上是由应用程序和 Windows 协同完成的。Windows 维护窗口的位置和显示,管理窗口的标准特征,如边框、滚动条和标题等,并进行许多初始化工作;应用程序维护有关窗口的其他工作,特别是负责维护窗口的“用户区域”(窗口边框之内的区域),对用户区域的显示进行完全的控制。

2. 事件驱动的机制

人机交互的理想方式就是要机器以一种自然的方式与用户交互。在传统的或“过程化”的应用程序中,应用程序自身控制了执行哪一部分代码和按何种顺序执行代码。从第一行代码执行程序并按应用程序中预定的路径执行,必要时调用过程,这是一种面向过程、顺序驱动的方式。程序必须有一个明确的开始、中间(每一时刻应该发生什么是预先定好的)及结束过程。用户也只能按照程序确定的流程被动地走下去。在事务越来越复杂、实时性要求越来越高的今天,顺序驱动的方式已不能满足要求了,事件驱动的机制能很好地解决这个问题。在事件驱动机制的应用程序中,代码不是按照预定的路径执行,而是在响应不同的事件时执行不同的代码片段。事件可以通过用户诸如单击鼠标或按下按键的操作触发、也可以由来自操作系统或其他应用程序的消息触发,甚至可以由应用程序本身的消息触发。这些事件的顺序决定了代码执行的顺序,因此应用程序每次运行时所经过的代码的路径都是不同的。显然,事件驱动的机制不由事先编好的顺序控制,而由事件发生的顺序控制,这样能更好地与现实世界接近。

Windows 正是采用了这种事件驱动的机制。而实现时,又是以消息为基础的;每个事件对应一个消息,应用程序围绕着消息的产生、分送、处理而展开。Windows 有三类基本的事件,它们是键盘事件(按下、放开……)、鼠标事件(单击、移动……)和时钟事件(一种系统时钟)。还有从基本事件或其他应用程序中派生出来的事件,比如关闭一个应用程序时,上一层窗口将收到由该事件产生的消息。

Windows 提供完善的事件驱动的机制,通过它用户能十分方便地实现对操作的响应。在事件驱动的机制中,任务的执行是对事件的响应,应用程序始终在运行中等待事件的发生,然后执行相关的操作。Windows 控制整个的事件循环,实现多任务的并行操作、多程序的同时启动运行。

3. 消息循环及处理机制

Windows 中的消息大致可分为两类:输入消息和窗口管理消息。输入消息对键盘和鼠标的输入作出反应。窗口管理消息是由对窗口进行的操作引发的,比如最大化、最小化或关闭窗口。还有控件(如命令按钮、复选框)也是一类特殊的窗口对象,对它们进行操作也将产生窗口管理消息。Windows 的消息循环及处理机制如下:

Windows 把所有接收来自键盘、鼠标和计时器的输入消息放到相应应用程序的“消息队列”里面。当应用程序准备检索输入时,只要从其消息队列里读取下一条消息。由于 Windows 应用程序通过消息队列来接收输入,所以 Windows 应用程序的主要特征是“消息循环”。消息循环从应用程序的消息队列中检索输入消息,并把它们发送给相应的窗口。每个窗口都有相应的窗口函数来处理键盘输入消息或窗口管理消息。图 1.2 说明了 Windows 和应用程序如何相互协作地处理键盘输入消息。当用户按下并放开键时,Windows 接收键盘输入并将其转化为键盘输入消息存入系统消息队列,然后把键盘输入消息从系统消息队列复制到应用程序消息队列中。应用程序的消息循环检索到键盘输入消息时,把它们翻译成 ANSI 字符 WM_CHAR,并把 WM_CHAR 消息和键盘消息发送给有关窗口函数。然后窗口函数使用 TextOut 函数在窗口的用户区域里显示字符。有一点要注意,就是消息循环不直接把消息发送给窗口函数,窗口函数也不直接把字符输出到应用程序窗口,二者都要通过 Windows 相应的模块,如图 1.2 所示。

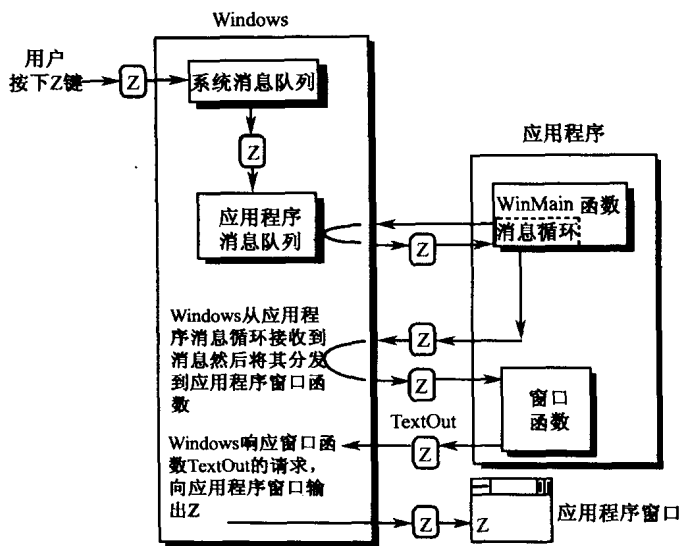


图 1.2 处理键盘输入

Windows 可以同时为多个应用程序接收和发送输入消息，如图 1.3 所示。Windows 收集所有输入操作，并以消息的形式存放到系统消息队列中，然后把每个输入消息复制到相应应用程序消息队列。每个应用程序的消息循环检索消息，通过 Windows 把它们发送到每个应用程序相应的窗口函数。

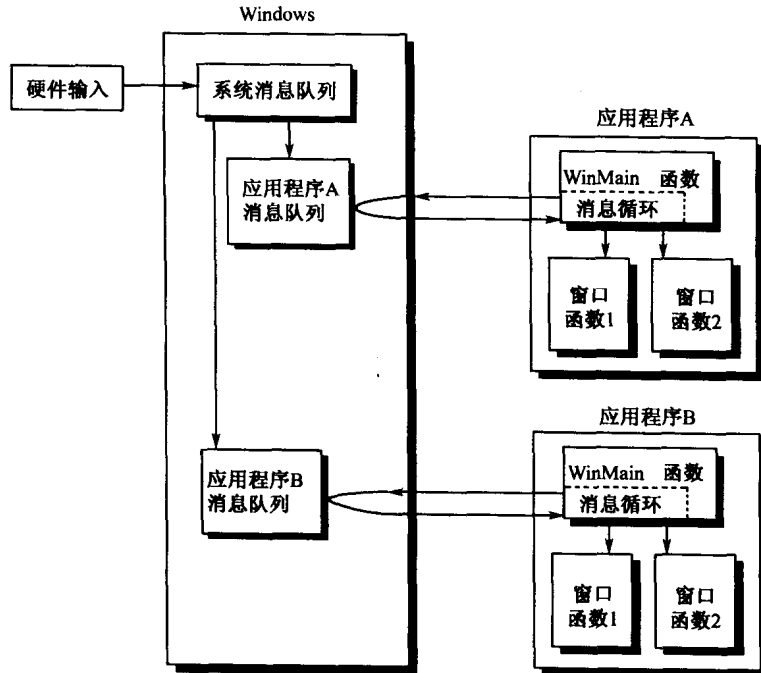


图 1.3 处理两个应用程序的输入