



北京市高等教育精品教材立项项目

高等学校教材

会计电算化系列



Internet

会计信息系统教程

● 王永生 编著 ● 陈维兴 主审



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

北京市高等教育精品教材立项项目
高等学校教材·会计电算化系列

Internet 会计信息系统教程

王永生 编著
陈维兴 主审

電子工業出版社·

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书是“高等学校教材会计电算化系列”之一,全面系统地讲述了当今面向 Internet 的会计信息系统基本概念、设计开发、运行维护以及发展前景。主要内容有 Windows 的机制、程序设计的基本概念、可视化编程、多媒体技术、会计信息系统的分析设计与开发、账务子系统的设计与开发、面向 Internet 会计软件的开发、会计决策支持系统等。

全书章节编排符合人的逻辑思维顺序,易于读者理解和记忆;阐述由浅入深,循序渐进;并穿插丰富的会计软件开发源程序实例(均在机上调试通过),利于读者操练和掌握所学内容。书中各章都配有适量的(上机)习题。

本书可作为大专院校财会专业“会计信息系统”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。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

Internet 会计信息系统教程/王永生编著. —北京:电子工业出版社, 2004.6

高等学校教材.会计电算化系列

ISBN 7-5053-9803-2

I. I… II. 王… III. 因特网—应用—会计—管理信息系统—高等学校—教材 IV. F232

中国版本图书馆 CIP 数据核字(2004)第 026565 号

责任编辑:王 颖

印 刷:北京牛山世兴印刷厂

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

经 销:各地新华书店

开 本:787×1092 1/16 印张:18.25 字数:468 千字

印 次:2004 年 6 月第 1 次印刷

印 数:5 000 册 定价:26.00 元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。联系电话:(010) 68279077。质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

前言

教材是教师授课取材之源,是学生求知学习之本。多年来作者一直注重创编教材,编了不少教材,获过多次奖励。此次北京市“精品教材立项”使作者有机会更好地编写普通高校大学本科会计电算化系列新教材。

一、编写目的

当今会计电算化工作(包括电算化软硬件、人才培养等)已开始进入 Internet 时代,与之相应的大学培养财会人才的教学工作,以及配套的会计电算化教材建设工作也迈进了一个新阶段,面临着开创性的紧迫任务。但是,目前财会专业会计电算化的教学水平与我国经济技术的发展、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。有的专业的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;有的专业的课程虽然分开上,但是没有系列教材,各课程教材内容不能很好地衔接,影响整体教学效果,因此要尽快建设会计电算化系列课程。若单独一门一门地建设,即便这些课程从各自的角度看都臻于完善,但其导致的后果很可能是限制了而不是有利于教学教改的深化发展,因此会计电算化系列课程之间要融会贯通,前后课程要互相衔接,形成新的会计电算化系列课程体系。这是搞好会计电算化教学教改、提高教学水平、紧跟时代发展的必然之路。

会计电算化教学内容要提高一个层次水平,使学生学到一定的深度与广度,毕业后在实际工作中才能应对自如,表现出很强的后劲,以适应会计电算化水平不断提高的需要。将来开发、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“自己教自己”。这也反映了当前会计电算化教学上存在的差距,同样会计电算化教材上也存在差距。

正是基于此,作者经过精心策划,运用多年教学、研究应用的经验,密切结合课程设置,创编了这套面向新世纪的、力争达国内一流水平的会计电算化系列新教材——《Internet 会计电算化基础》、《Internet 电算化会计实务》和《Internet 会计信息系统教程》。这三本新教材从整体上体现了新世纪财会人员应具备的会计电算化知识技能结构体系,从内容上反映了国内外这方面的最新成果和会计电算化的发展趋势。各书之间融会贯通,既相互衔接又不重复,知识结构更加合理,内容与新世纪需求更相适应,能很好地满足 Internet 环境下会计电算化教学发展的迫切需要。

作者在多年的会计电算化教学工作中,已经开设了会计电算化系列课程即“会计电算化基础”、“会计信息系统”和“电算化会计实务”三门课程,受到了学生的好评。加之上机实践,收效事半功倍。特别在会计学专业本科班上开设“会计信息系统”课程,增加会计软件开发技术的内容,让学生模仿真实开发会计软件,达到一定的深度与广度,学生普遍感到会计电算化水平大有提高,以至于在学生就业时起到了很大的作用。教学改革实践证明,会计电算化教学内容要提高一个层次水平,首先要有高水平的教材,本系列教材将发挥很好的作用。

二、教材特点

(1) 本系列教材是本科会计学专业会计电算化的基础课、专业基础课和专业主干课的系

列配套教材, 又是文理渗透、学科交叉、在内容体系等方面具有创新特点的教改教材。

(2) 本系列教材处理好了系列课程中前、后续课程的诸多内容协调衔接关系, 确定好了系列教材内容的取舍、侧重点和融会贯通之处。内容定位准确, 符合时代需求, 有利于财会学生当前学习与未来发展。

(3) 本系列教材拓宽基础、加强实用, 改变本科会计学专业教学滞后于科技、经济、社会发展的现有状况, 强化培养学生既懂会计又懂计算机的财会技能, 使学生适应新世纪对人才知识结构的新要求, 成为深受社会欢迎的、具有创新能力、自学能力、分析解决实际问题能力、电算化会计实际动手能力与组织管理能力的高素质复合型财会人才。

(4) 全套书内容取材新颖, 引人入胜; 编排层次分明, 逻辑性强, 文字、图形、程序并茂; 阐述由浅入深, 由表及里, 通俗易懂, 符合循序渐进的认识规律, 便于自学; 具有系统性与完整性、科学性与实用性、知识性与趣味性、理论性与实践性。

(5) 书中介绍原理、讲解方法、操作功能、讨论问题, 均结合实例进行, 这样给学习者创造一个轻松快捷的学习、仿效、练习和掌握会计电算化知识与技能的最佳学习环境。章节小标题也突出实用性, 这样遇到问题时, 马上就能找到所需答案, 从而使问题快速得到解决。赢得时间, 赢得效益。

三、学习提示

(1) 学习者应具备的基础。学习者应具备计算机操作的基本知识, 最好在读完《Internet 电算化会计基础》之后开始本书学习, 这样会减少学习困难, 收效更佳。

(2) 学习与实践相结合。学习书上内容, 要注重实践。要仿效、练习书中的例子, 独立完成习题作业。要多动脑思考, 追究其根源。要多联系记忆, 勿盲目操作。这样边学习新知识, 边上机实践, 才能很好地掌握所学内容。

(3) 遇到学习困难找“帮助”。在学习、上机过程中, 会遇到一些问题, 作者提倡学习者自己解决。可取的方法是, 请教软件的帮助系统, 这样既可以解决所遇到的问题又可以更熟悉软件, 增加感性认识, 对学习书中内容大有益处, 久而久之, 便无书自通。

(4) 上机以前多预习。上机时间有限, 应在上机以前多预习, 做好准备工作。做到上机时心里有数, 把操作步骤一步步装在脑子里或写在纸上, 照着练习即可。

(5) 要有坚持不懈的精神。学习会计电算化系列教材需要较长的时间, 不可能一帆风顺, 因此要有坚持不懈的精神。

本书是会计电算化系列新教材之一, 全书共分 10 章。第 1 章阐述 Windows 环境与机制; 第 2 章讲述程序设计基本概念; 第 3 章讨论可视化编程; 第 4 章讲解控件使用; 第 5 章介绍多媒体技术; 第 6 章概述会计信息系统及其发展前景; 第 7 章介绍会计信息系统分析与设计方法; 第 8 章讨论账务子系统设计与开发; 第 9 章讨论面向 Internet 会计软件开发; 第 10 章介绍会计决策支持系统。书中各章均配有适量练习与上机题。学时为 60 左右, 上机学时约占 40%~50%。最好是边上机讲课, 边上机练习。

本书由王永生教授编著, 参加相关工作的还有黄平生、艾志刚、叶小玲、徐颖等。

全书由陈维兴教授主审。

书中难免有错, 敬请读者指正。

编著者

2003 年 12 月

目 录

第 1 章 Windows 环境与机制简介	(1)
1.1 Windows 环境	(1)
1.1.1 图形用户界面 (GUI)	(2)
1.1.2 图形设备接口 (GDI)	(2)
1.1.3 多任务处理	(2)
1.1.4 信息交换	(3)
1.1.5 开放数据库链接 (ODBC)	(4)
1.1.6 内存管理	(4)
1.1.7 真型 (TrueType) 字体技术	(4)
1.1.8 多媒体技术	(4)
1.1.9 网络支持	(5)
1.2 Windows 机制	(5)
1.2.1 控件及对象的概念	(5)
1.2.2 事件驱动的机制	(5)
1.2.3 消息循环及处理机制	(6)
1.3 Windows 开发工具	(9)
1.3.1 元件开发工具 (Component Builder)	(9)
1.3.2 解决方案开发工具 (Solution Builder)	(9)
1.3.3 决策支持及查询工具 (Decision Support and Query Tools)	(10)
习题 1	(10)
第 2 章 程序设计的基本概念	(11)
2.1 程序设计的优点	(11)
2.2 程序设计的基本概念	(12)
2.2.1 存储数据	(12)
2.2.2 处理数据	(13)
2.2.3 自定义函数和过程	(15)
2.2.4 程序流控制	(18)
2.3 面向对象程序设计的基本概念	(20)
2.3.1 对象	(20)
2.3.2 类	(22)
2.3.3 事件	(22)
2.3.4 控件	(23)
2.3.5 容器	(23)
2.3.6 对象的属性	(23)
2.3.7 对象的方法	(24)

2.3.8 对象的事件	(24)
2.3.9 对象引用	(26)
2.3.10 设置属性	(28)
2.3.11 响应事件	(28)
2.3.12 调用方法	(31)
习题 2	(31)
第 3 章 可视化编程	(32)
3.1 画各种控件	(32)
3.1.1 画控件	(32)
3.1.2 各种控件	(35)
3.2 编程机制	(46)
3.2.1 使用“命令”窗口	(46)
3.2.2 创建程序	(46)
3.2.3 编写代码	(47)
3.3 可视化编程	(48)
3.3.1 布局、设置对象属性	(48)
3.3.2 控制逻辑与何地代码	(51)
3.3.3 编程示例	(54)
习题 3	(58)
第 4 章 控件使用	(59)
4.1 深入了解控件和数据	(59)
4.1.1 控件和数据的关系	(59)
4.1.2 控制对数据的访问	(60)
4.2 根据任务使用控件	(66)
4.2.1 使用命令控件	(66)
4.2.2 使用选项控件	(67)
4.2.3 使用输入控件	(74)
4.2.4 使用微调控件	(77)
4.2.5 使用显示控件	(77)
4.2.6 使用时间控件	(80)
4.3 使控件易于使用	(81)
4.3.1 设置访问键	(82)
4.3.2 设置控件的 Tab 键次序	(82)
4.3.3 设置工具提示文本	(82)
4.3.4 更改鼠标指针显示	(83)
4.3.5 启用和废止控件	(83)
4.3.6 允许用户拖放	(83)
4.4 创建任务控件	(83)
4.4.1 决定什么情况下创建控件	(84)
4.4.2 创建任务控件实例	(84)

4.4.3	修改控件	(102)
4.4.4	复制、删除或查看类	(103)
4.4.5	以编程方式定义类	(104)
习题 4		(105)
第 5 章	多媒体技术	(106)
5.1	多媒体编程技术	(106)
5.1.1	媒体控制接口	(106)
5.1.2	Windows 应用程序编程接口	(110)
5.1.3	VF 调用动态链接库函数	(110)
5.1.4	调用多媒体函数 mciSendString	(112)
5.1.5	VF 多媒体编程	(113)
5.2	自制声音图标按钮控件	(117)
5.2.1	定义声音图标按钮控件类	(118)
5.2.2	定制声音图标按钮控件	(122)
5.2.3	使用声音图标按钮控件	(124)
习题 5		(125)
第 6 章	会计信息系统概述	(126)
6.1	何谓会计信息系统	(126)
6.1.1	会计信息	(126)
6.1.2	会计信息系统	(130)
6.1.3	计算机会计信息系统	(133)
6.2	会计信息系统核算制度	(135)
6.2.1	记账方法的规范化	(135)
6.2.2	账户凭证账簿报表的设置	(138)
6.2.3	账务处理的基本模式	(145)
6.3	会计信息系统的发展前景	(151)
习题 6		(153)
第 7 章	会计信息系统开发方法	(154)
7.1	系统开发方法	(154)
7.1.1	原型法	(154)
7.1.2	面向对象法	(156)
7.1.3	生命周期法	(158)
7.2	会计信息系统的总体分析	(160)
7.2.1	系统目标与规模确定	(160)
7.2.2	可行性分析	(160)
7.2.3	业务调查	(161)
7.2.4	系统的子系统划分	(163)
7.2.5	数据流图	(165)
7.2.6	数据词典	(170)
7.2.7	系统分析报告	(176)

7.3	会计信息系统的总体设计	(177)
7.3.1	会计信息系统的设计原则	(177)
7.3.2	系统网络设计	(179)
7.3.3	系统数据库设计	(184)
7.3.4	代码设计	(186)
7.3.5	输入输出设计	(187)
7.3.6	系统设计说明书	(188)
7.4	会计信息系统的实施	(188)
7.4.1	可视化编程	(188)
7.4.2	系统实施	(190)
7.5	会计信息系统的维护	(192)
7.5.1	系统评审与评价	(192)
7.5.2	系统使用	(193)
7.5.3	系统维护	(193)
习题 7		(194)
第 8 章	账务子系统的设计与开发	(195)
8.1	会计科目体系维护功能的设计与开发	(195)
8.1.1	为什么要对科目进行编码	(195)
8.1.2	科目编码方式与编码结构	(196)
8.1.3	科目代码结构的灵活性与助记码	(197)
8.1.4	校验码的设置	(197)
8.1.5	会计科目体系维护功能的设计	(198)
8.1.6	会计科目体系维护功能的可视化编程	(199)
8.2	凭证处理子系统的总体设计	(223)
8.2.1	凭证子系统的功能分析	(223)
8.2.2	凭证子系统数据库的设计	(225)
8.3	凭证输入修改模块的设计与开发	(228)
8.3.1	凭证输入格式的设计	(228)
8.3.2	凭证输入控制的设计	(230)
8.3.3	凭证输入修改模块功能的设计	(232)
8.4	凭证审核模块的设计与开发	(235)
8.5	凭证汇总模块的设计与开发	(236)
8.5.1	凭证汇总的功能分析	(236)
8.5.2	凭证汇总模块的功能设计	(236)
8.6	凭证查询模块的设计与开发	(237)
8.6.1	凭证查询的功能分析	(237)
8.6.2	凭证查询模块的功能设计	(237)
习题 8		(238)
第 9 章	面向 Internet 会计软件的开发	(240)
9.1	会计软件的开发	(240)

9.1.1	解决问题的分析	(240)
9.1.2	创建数据库	(241)
9.1.3	可视化编程	(241)
9.1.4	软件的测试与调试	(242)
9.2	开发多用户应用程序	(243)
9.2.1	相关术语	(243)
9.2.2	主要问题	(244)
9.2.3	编写多用户应用程序	(244)
9.3	开发 C/S 应用程序	(246)
9.4	Socket 接口	(252)
9.5	会计软件连编	(253)
9.5.1	会计软件项目构成	(253)
9.5.2	使用项目管理器	(256)
9.5.3	连编会计软件项目	(258)
9.5.4	运行会计软件	(259)
9.6	会计软件发布	(259)
9.6.1	准备发布文件	(260)
9.6.2	建立发布树	(261)
9.6.3	创建发布磁盘	(262)
习题 9		(268)
第 10 章	会计决策支持系统	(269)
10.1	会计决策支持系统的产生	(269)
10.1.1	现代会计职能要求参与决策	(269)
10.1.2	会计决策支持系统开发势在必行	(270)
10.1.3	会计电算化发展的高级阶段	(271)
10.2	决策支持系统的组成	(272)
10.2.1	模型库	(272)
10.2.2	方法库	(272)
10.2.3	数据库	(273)
10.2.4	人机界面	(274)
10.3	会计决策的内容和模型	(275)
10.3.1	会计决策的内容	(275)
10.3.2	会计决策模型	(275)
10.3.3	常用的方法	(276)
10.4	会计决策支持系统的开发方法	(277)
10.5	会计决策支持系统举例	(278)
习题 10		(280)

第 1 章 Windows 环境与机制简介

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。本章将对这些内容进行较深入的讨论，为进一步学习会计信息系统开发方法与编程调试，奠定良好的基础。

1.1 Windows 环境

Windows 环境是指它提供的资源、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，等等。

在 Windows 环境中，资源不仅包括系统的硬件资源，如显示卡、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、字符串等都能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 内存不足时还可以放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，故容易将应用程序做本地化。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、链接即可。不必逐行查找和修改代码。

有人说，Windows 犹如一株人参，如图 1.1 所示。

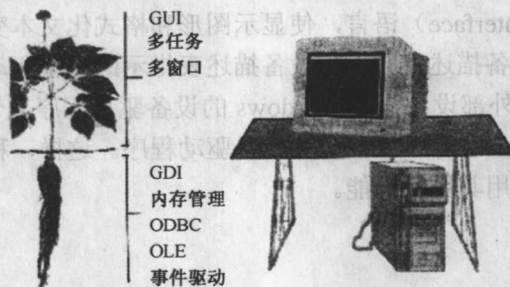


图 1.1 Windows 是一株人参

对最终用户来讲，Windows 呈现的只是地面之上漂亮的花朵。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、界面新颖美观的应用程序，这对 Windows 本身也是增值。Windows 程序员最能懂得“人参”地面之下最有价值的东西，懂得怎样才能充分发挥它的功能。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

图形用户界面以及与设备无关的图形设备接口（GDI）已成为 PC 事实上的操作系统标准。

1.1.1 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、输出也变得简单、高效。屏幕从 DOS 中键入命令或文本信息的字符方式中解脱出来，向用户提供了直接、丰富、自然的交互手段。GUI 的作用不止是从 DOS 的字符环境走入了图形环境。作为 PC 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像，是多媒体软件开发的理想界面。

1.1.2 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，所以可以使用同一函数在打印机或高分辨率显示器上画同一个图。编写 Windows 的应用程序，不需要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。Windows 为此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。

1.1.3 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和输入、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同时处理多个任务，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片来访问 CPU。由于 CPU 切换速度极快，使人们感觉是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。即一个应用程序使用完 CPU 后释放, 另一个应用程序再使用再释放, 依次进行。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, 则多项任务将被迫中止, 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由 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。

1.1.4 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, 放到“剪贴板”上, 然后再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。用户可以利用 DLL 中的函数和资源来建立 Windows 应用程序, 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 DLL 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 的字体实际上是可以通过 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。产生对象的文档称做源文档, 放入对象的文档称做目标文档。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映到源文档中。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。

1.1.5 开放数据库链接 (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。在他们忙着争夺市场时, 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连, 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, 给开发和维护工作带来了很大困难。数据库互连的开放化、标准化势在必行。

1991 年 11 月, 微软宣布了 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, 不必了解内部的机理。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。

1.1.6 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, 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。

1.1.7 真型 (TrueType) 字体技术

真型字体技术即 TrueType 字体技术。计算机系统输出设备主要有: 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 (What You See Is What You Get. WYSIWYG)。

1.1.8 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、动画、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。

1.1.9 网络支持

Windows 是在 PC 上发展起来的。20 世纪 80 至 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，并成为当今信息技术的热点。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。而到了 Windows 95，很大程度上是一个网络操作系统，特别是 Windows XP，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。这使得 Visual FoxPro 可支持客户/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。

1.2.1 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，它不仅用做输出设备，还可以用做输入设备，就像真实设备的控制界面。这种显示在应用程序的图形用户界面中，可供用户操纵并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。特别地，应用程序负责维护窗口的“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控制。

1.2.2 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的那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。事件驱动的机制很好地解决了这个问题。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，执行程序代码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（按下键、放开键……）、鼠标事件（单击鼠标、移动鼠标……）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 提供了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。

1.2.3 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做出反应。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”中。当应用程序准备检索输入时，只需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 Windows 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。

Windows 可以同时为多个应用程序接收和分送输入消息，如图 1.3 所示。Windows 收集所有输入，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的应用程序消息队列。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。

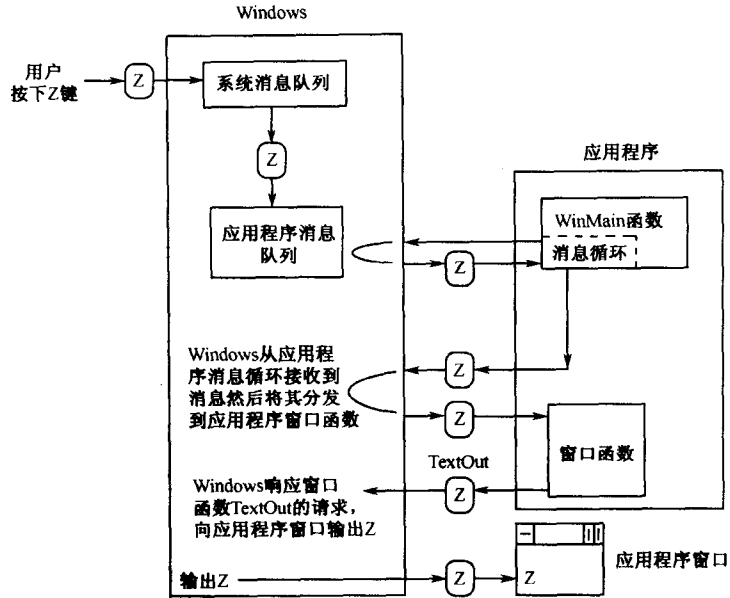


图 1.2 处理键盘输入消息

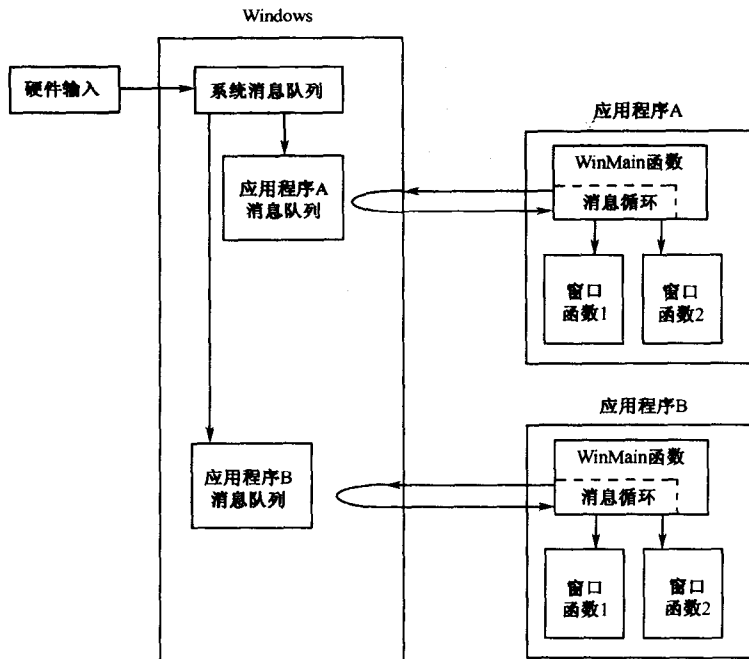


图 1.3 处理两个应用程序的输入

与输入消息不同（它必须经过应用程序的消息队列），Windows 直接把窗口管理消息发送给有关窗口函数。图 1.4 说明了 Windows 如何把窗口管理消息发送给窗口函数。在 Windows 接收到删除应用程序窗口的请求之后，它跳过应用程序消息队列直接把 WM_DESTROY 消息