

FOXPRO

事件驱动程序设计



- Explains the essentials for DOS & Windows database application developers
- Expert advice & programming answers
- Master strategies to help you tackle the most demanding tasks



*Disk Includes
Sample Code from
Book Examples
& Full Contact
Manager Demo!*

[美]

Dick Bard

著

赵 军 李 杜
成 昊

译
审校

清华大学出版社

Brady

北京科海培训中心

FoxPro 事件驱动程序设计

[美] Dick Bard 著

赵军 李杜 译

成昊 审校

清华大学出版社

FoxPro 事件驱动程序设计

FoxPro Event-Driven Programming

Copyright © 1993 by Brady Publishing

All rights reserved. No part of this book shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from the publisher.

本书英文版由 Prentice Hall 出版社属下的 Brady 计算机图书出版公司于 1993 年出版。版权为 Brady 公司所有。本书的中文版版权由 Prentice Hall 公司授予北京科海培训中心和清华大学出版社合作共同出版、发行。未经出版者书面允许不得以任何方式复制或抄袭本书内容。

(京)新登字 158 号

FoxPro 事件驱动程序设计

[美] Dick Bard 著
赵军 李杜 译
成 昊 审校

☆

清华大学出版社出版

北京 清华园

门头沟胶印厂印刷

新华书店总店科技发行所发行

☆

开本: 787×1092 1/16 印张: 25.5 字数: 620 千字

1994 年 9 月第 1 版 1994 年 9 月第 1 次印刷

印数: 0001—8000 册

ISBN 7-302-01658-5/TP·708

定价: 50.00 元

序 言

我坚信 Dick 编写的《FoxPro 事件驱动程序设计》会成为一本畅销书。Dick 使用自己创新的方法设计和编写事件驱动程序,他在这方面已久负胜名。看到他把自己的经验和技術奉獻給廣大 FoxPro 程序员和用戶,我感到由衷的欣慰。

《FoxPro 事件驱动程序设计》一书集中了大量 FoxPro 优秀书籍中的高级开发技术。大多数 FoxPro 程序设计方面的书都集中介绍各自的专题。Dick 的这本书集中介绍了开发事件驱动接口程序的技术,这种技术对于大多数 FoxPro 程序员都是陌生的。阅读本书之后,给我印象最深的是 Dick 为事件驱动程序设计提供了一套系统的方法——对于我来说,这也许是本书最有价值的部分。Dick 提供的示例程序也是极有价值的,它们在本书中充分说明了作者如何设计那些接口程序(或界面)。

FoxPro 为开发复杂的界面提供了功能强大的工具,但要充分使用它们,使用者必须具有一定的编程技能和经验。然而,《FoxPro 事件驱动程序设计》将为你提供获取这些技能和经验的捷径,每个用 FoxPro 编写事件驱动程序的人都会从中受益。

Walter Kennamer

微软公司软件设计工程师

引言

用户首次使用 FoxPro 就可能为之倾倒。用户可以根据需要同时打开多个窗口,并可随意移动它们——所有这一切只需按动鼠标。

数据库领域正在发生重大变革, FoxPro 允许我们在管理数据时使用窗口。FoxPro 让我们控制自己的计算机,并以不同的方式在我们需要信息的任何时刻组织它们——在不同的窗口中把它们组织在一起。

FoxPro 的功能不仅仅只局限于这花哨的表面。通过把我们需要的信息一块块组织起来,我们就能发现信息之间的总体联系——非程序员甚至也能看到这种联系,并提供常规的固定屏幕的应用程序。把信息划分到不同窗口中,然后把这些窗口集中在一起,我们就能更高效地使用数据库了。

FoxPro 的事件驱动程序设计能力允许程序员把这种多窗口的功能赋予用之编写的应用程序。这些应用程序管理的信息可以是部分客户通讯录管理系统、顾客管理系统、物资票据管理系统、桌面帮助(Help Desk)系统、销售系统——它的信息可以涉及顾客追踪,各种推销手段,个人管理,等等。我们可以通过多窗口系统的透镜观察相关数据的新天地。

事件驱动系统极易使用。我们在使用字处理软件、电子表格软件以及使用诸如 Windows 或 DESQview 这类多任务操作系统的外壳时已经感受到这一点。

事件驱动的数据库应用程序同样易于使用。经理、事务管理程序员、专业人员等决策者,他们只需用鼠标即可完成复杂的任务,而不要求具有计算机的专业知识。

从我第一次使用 FoxPro 开始,就认为 FoxPro 最令人瞩目的特点是:多窗口程序设计功能和 FoxPro 的速度,同 FoxPro 奇快的速度一样,多窗口程序设计给应用程序开发者带来了许多机会。

众多 FoxPro 程序员已开始发出他们的心声:“必须有一些编程方法,使得编写的应用程序易于使用!”读者也许已经试着构造了类似于 FoxPro 的应用程序,也许已经了解到事件驱动接口程序使得用户使用其他软件更加容易。也许读者还不了解什么是事件驱动程序设计,那么就请加入我们的行列。

本书的内容

本书将教会读者构造多窗口应用程序的方法。读者可以使用本书配套磁盘上的示例程序(AYCDEMO.APP)作为入门教材。

提示: 配套磁盘上还含有一个示例程序(SAMPLE.PRg),其中含有基本的事件驱动系统的所有代码。读者在阅读本书的同时运行示例程序有利于读者理解本书的内容。

本书中的程序假定 FoxPro 的基本版本为 2.5。FoxPro for Windows 和 FoxPro for DOS 共享这个版本中的公用程序,这表明我们可以编写运行在这两种平台上的应用程序。当然,

有一些例外,但是它们不影响基本的事件驱动程序设计。

读者需要 FoxPro 2.5 和鼠标来运行演示程序或示例程序(需要 386 或 486 计算机来运行演示程序)。如果读者想在 FoxPro 2.0 以下版本中运行应用程序,则必须将一些 FoxPro 2.5 的命令注释掉。AYCDEMO.APP 和 SAMPLE.PRg 将运行在 FoxPro for Windows 下,但屏幕不重画为图形环境。如果在 FoxPro for DOS 下运行它们,将得到最佳的结果(可以在 DOS 和 Windows 中运行 FoxPro for DOS)。示例程序内指令的有关细节,参阅附录 A“基本系统”或附录 B“运行演示应用程序”。

本书以章节为序。第一部分总述事件驱动程序设计主要的问题以及编写事件驱动系统的一般方法,大多数的概念在这部分讲述。第二部分和第三部分把这些概念实现于程序中。附录 D 含有追加的源程序清单,本书配套磁盘上的程序文件含有全部源代码。

Windows 世界

在事件驱动应用程序中,用户可以先暂停他正在做的工作,阅读如下显示屏后再重新回到自己的工作(见图 I.1 和 I.2)。

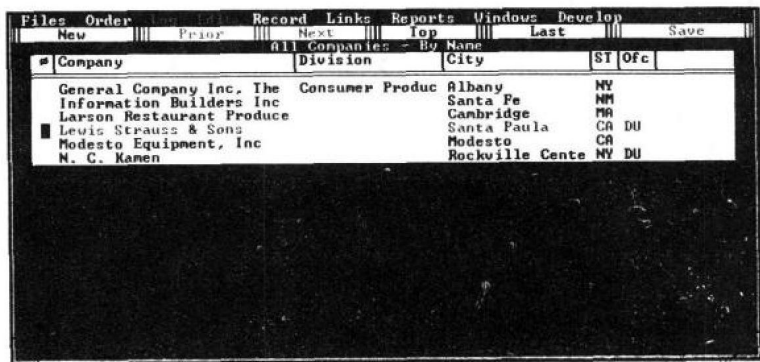


图 I.1 打开一个窗口...

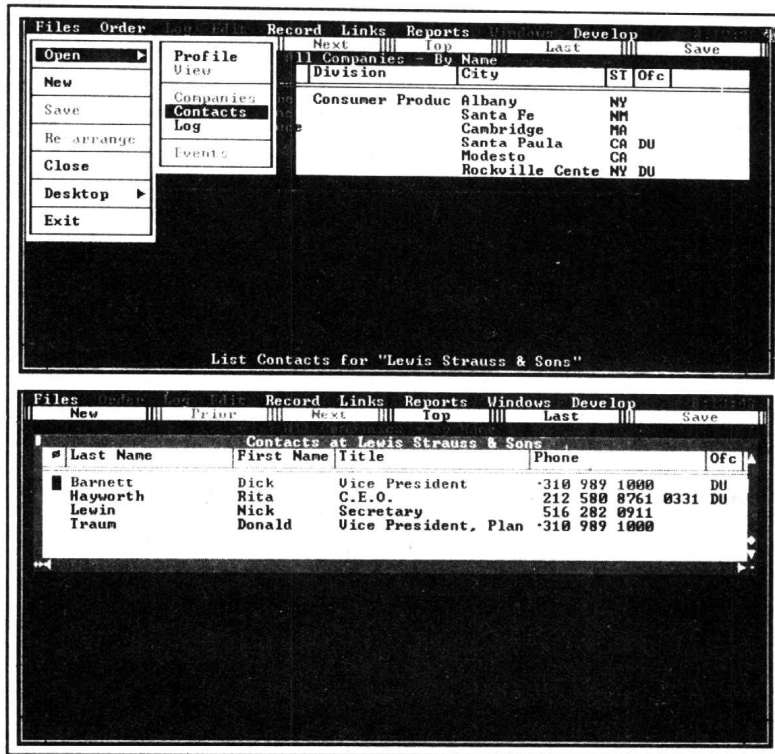


图 I. 2 和 I. 3 ...或打开两个窗口...

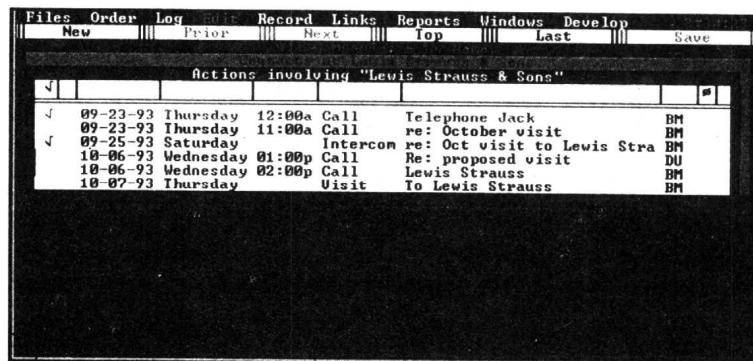


图 I. 4 ...或打开三个窗口...

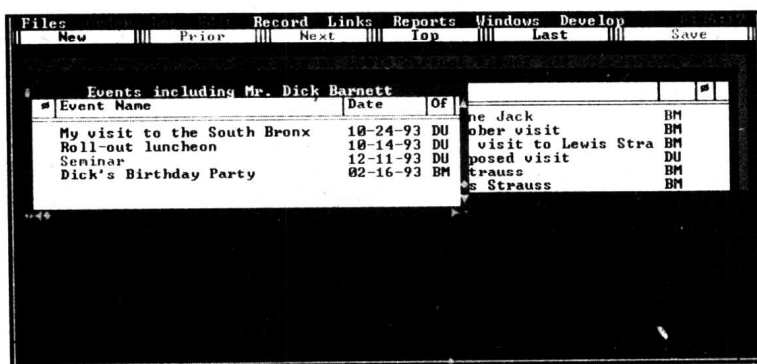


图 I.5 ...根据需要打开多个窗口...

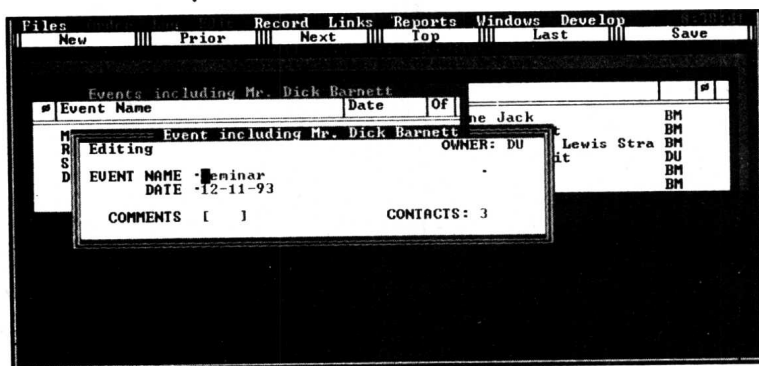


图 I.6 编辑任意文件中的记录(所需做的只是单击鼠标)...

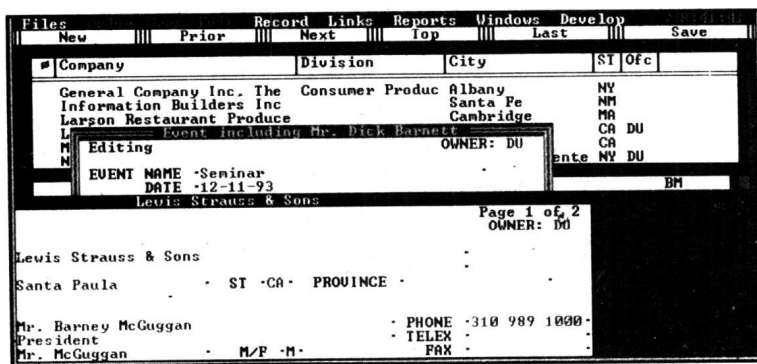


图 I.7 ...或者同时编辑两个文件中的记录...

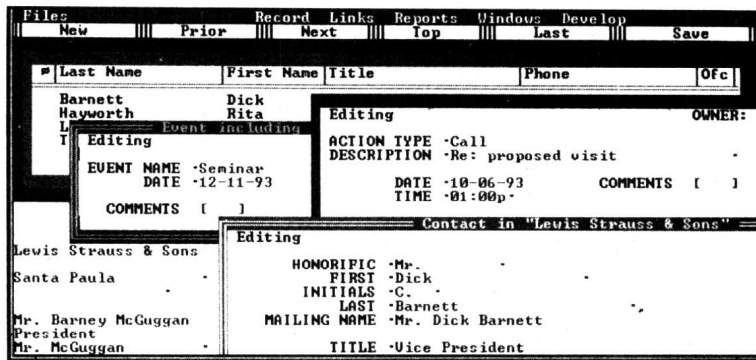


图 1.8 ...或者同时编辑三个或四个文件中的记录,或根据需要而定

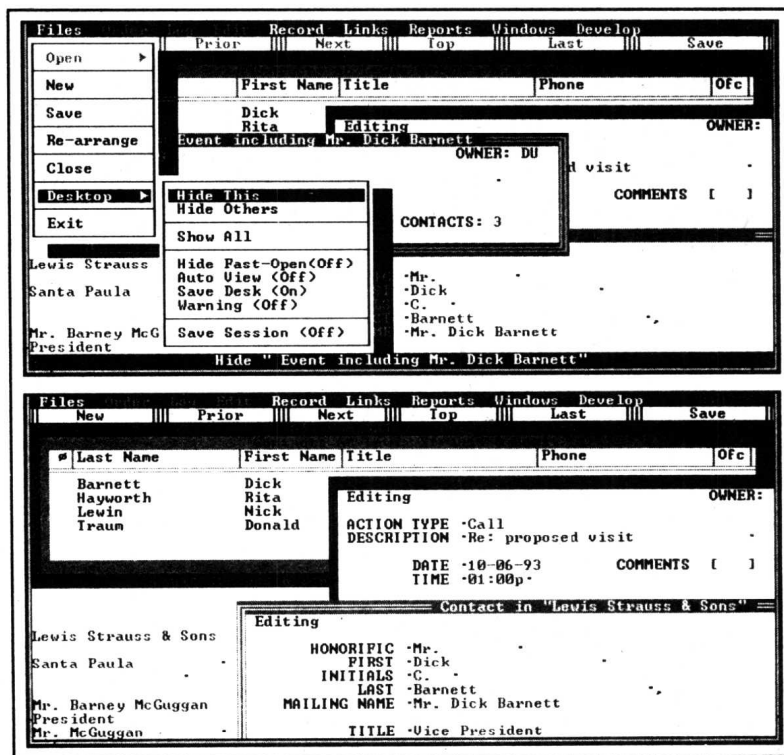


图 I.9 和 I.10 可以一次隐藏一个窗口

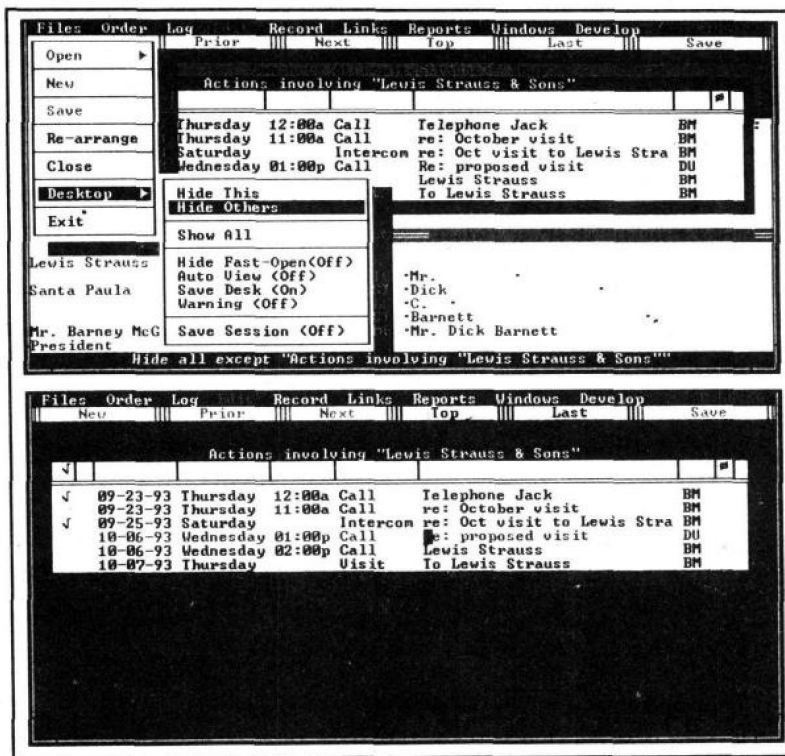


图 1.11 和 1.12 ...或者一同隐藏窗口

用户需要的方法

对于用户而言,不同窗口内的操作好像都在同时执行。然而,作为程序员,必须一次管理一个窗口:一个编辑窗口或一个浏览(BROWSE)窗口——这取决于用户当前正在使用的窗口。这和单窗口系统中的工作方式相同。

如果读者已编写过多文件、关系型应用程序,或者至少了解它们的工作原理,则可以开发多窗口系统。这些系统的一些基本原理同单窗口应用程序中的大多数原理类似。

多窗口系统还为程序员提供了各种选择的自由。构造一个系统不存在捷径,也不存在定义事件驱动程序设计的功能配置。

多窗口应用程序的各个窗口可相互链接(例如,Contact 窗口可以只显示某个公司的职员,而这个公司是 Companies 窗口内的当前记录)。各个窗口也可以处理彼此完全不相关的信息。

- 可以把窗口定义在用户可以移动的地方。
- 可以根据用户的愿望打开(或关闭)固定的一组窗口。
- 可以在窗口中作了一些改动时自动更新窗口,或者自动关闭它们(可以让用户自行选择)。

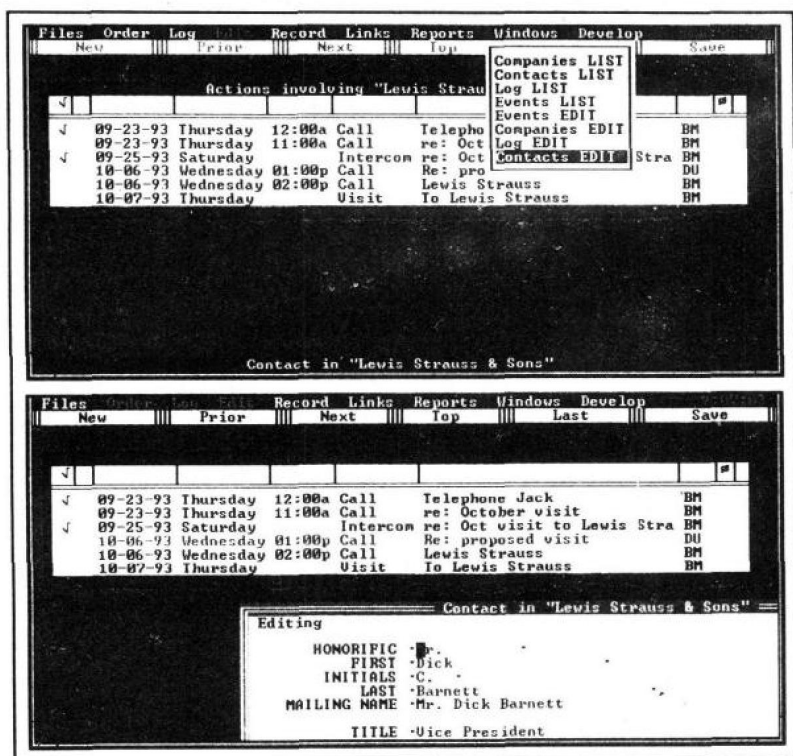


图 1.13 和 1.14 或者有选择地重新打开它们...

- 可以首先显示 READ 窗口或 BROWSE 窗口。任何时候,均可使用浏览窗口 (BROWSE)。
- 允许用户在 BROWSE 窗口内更换记录,或者禁止在 READ 窗口中编辑记录。
- 可以在用户退出 READ 窗口之前要求他们保存编辑过的内容。
- 在控制面板或工具栏菜单上加入按钮。
- 系统可以运行 SQL,它们也可以是基于 DBF 的,并支持活动记录。
- 系统可以是固定编码的,或者它们是可重配置的和数据驱动的。
- 系统提供一组固定的菜单选项,这些选项反映系统实现的功能。
- 系统让用户在进行下一个会话前或者在每次会话开始时保存所做的工作。
- 系统可以支持多个用户或者一个用户。

存在一些可替代的功能,但是它们都不是可有可无的。我们可以综合这些功能(例如,某些 READ 规定执行的方式而其他则不是),可以支持这些功能的任意一种组合方式。

我们甚至可以把自己的多窗口系统调整到局部内存的范围内,不同版本的 FoxPro 具有不同数量的工作区。

本书的任务之一就是教会用户构造一个事件驱动系统,让用户在自己的多窗口应用程序中加入自己所需的各种功能。在学习这种技术之前,我们首先看看在事件驱动系统之前应该掌握什么,以及为什么要掌握它们。

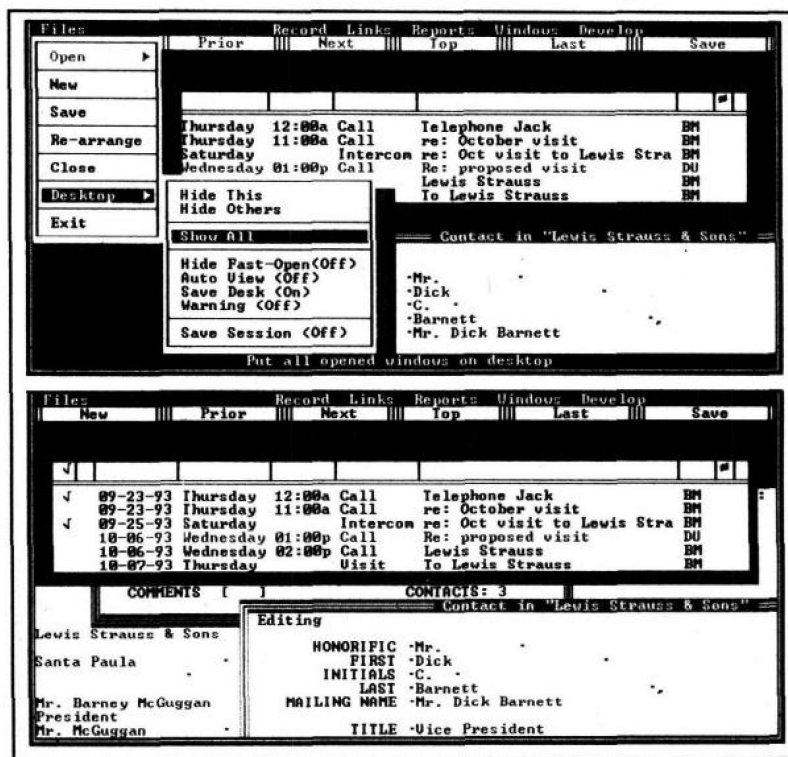


图 1.15 和 1.16 ...或者同时打开这些窗口

多窗口程序设计

任何窗口都代表一个进程,一个 BROWSE 或 READ(以及正在处理的文件)。在多窗口系统中,用户不但可以随时打开新的窗口来启动新的进程,而且具有重大意义的是:用户可以返回到以前打开的任意一个窗口,只要用户在打开这个窗口之后并未关闭它。

看待这一过程的另一种方法是,用户可以重新进入进程链上的任意一个进程,所做的工作只是用鼠标单击进程对应的窗口。

本书主要讨论如何管理不同窗口中运行的 BROWSE 和 READ 进程——运行时的情况如何。

多窗口程序设计最基本的要求是:对于用户所发出的动作,应用程序都必须知道:

- 用户目前在哪个窗口内工作
- 用户什么时候决定做其他工作
- 其他工作是什么

一旦知道用户正在工作的窗口,处理这个窗口的程序就完全同以前使用的程序相同。浏览(BROWSE)这个文件,编辑(READ)相应的记录。

如果用户掌握了这一方法,编辑就迎刃而解了。

用事件传送信息

事件驱动程序设计要求我们标识用户发出的任何动作:例如,从一个窗口移到另一个窗口,键入一个热键,选择菜单。这些动作就是事件(event)。通过把有关事件的信息转化为消息(message),使事件信息清晰、明确。

然后使用这些消息通知执行用户动作的应用程序。

可以使用事件消息(event message)告诉应用程序用户想实现的操作,无论用户何时发出他已准备进行工作的信号:

- SWAPWIN:运行进行窗口切换的例程。
- CLOSE:运行关闭窗口的例程(并切换到另一个窗口)。
- EDWINDOW:调用启动 READ 窗口的例程,使之编辑 BROWSE 窗口中的当前记录。
- MENU:调用处理菜单的例程。
- NEW:调用增加一条新记录的例程。
- PRINT:运行 Report 子系统的例程。

本书讲述的事件驱动程序设计技术也可以应用于单窗口系统。更重要的是,它为我们提供了管理多窗口系统的基础。

多窗口程序设计是基本事件处理技术的扩充,它允许我们追踪用户的工作点——当用户启动一个新事件时追踪依旧继续。

虽然多窗口程序设计确实比单窗口程序设计复杂,但是一旦掌握它,编写多窗口应用程序会更有趣。

目 录

第一部分 基本方法

引 言	(1)
第1章 人 门	(1)
1.1 入 门	(1)
1.1.1 决定事件驱动接口程序的功能	(4)
1.1.2 设计思想的一个例子	(4)
1.2 采用一种通用的方法	(6)
1.2.1 在应用程序中加入事件驱动功能	(6)
1.2.2 了解用户需求	(6)
1.2.3 多窗口管理	(7)
1.3 使用基于消息的方法	(10)
1.4 信 息 管 理	(10)
1.5 保持清晰的信息流	(11)
1.6 后续章节的内容	(12)
第2章 事件管理	(13)
2.1 产生事件	(14)
2.1.1 窗口转换事件	(15)
2.1.2 热键事件	(15)
2.1.3 菜单事件与按钮菜单	(16)
2.2 捕 获 事 件	(19)
2.3 使用事件捕获函数	(19)
2.3.1 第1个问题:窗口是否已关闭?	(21)
2.3.2 第2个问题:是否已选择了一个菜单子项?	(22)
2.3.3 第3个问题:是否已选择了一个菜单按钮?	(23)
2.3.4 第4个问题:是否按了热键?	(23)
2.3.5 第5个问题:用户是否选择了另一窗口?	(24)
2.3.6 第6个问题:是否发生超时事件?	(24)
2.3.7 捕获记录更换事件	(25)
2.3.8 终止全体事件的 BROWSE 或 READ 会话	(26)
2.3.9 BROWSE 与 READ 的差异	(26)
2.3.10 以同样方式处理 BROWSE 和 READ	(27)
2.4 事件处理	(27)

2.4.1	事件处理程序产生事件.....	(29)
2.4.2	了解窗口事件.....	(29)
2.5	组合 BROWSE 和 READ	(29)
2.6	READ 基本处理	(31)
第3章	菜单管理	(34)
3.1	了解菜单和菜单控制规则.....	(34)
3.1.1	事务控制规则.....	(34)
3.1.2	逻辑控制规则.....	(35)
3.1.3	系统控制规则.....	(35)
3.1.4	接口控制规则.....	(35)
3.1.5	组合菜单控制规则.....	(35)
3.2	执行菜单控制规则.....	(36)
3.2.1	创建菜单系统.....	(38)
3.3	设置一个菜单系统.....	(39)
3.3.1	提供对菜单的访问.....	(39)
3.3.1.1	定义菜单子项.....	(39)
3.3.1.2	重置菜单子项标志.....	(42)
3.3.2	产生菜单事件.....	(44)
3.3.3	定义弹出子菜单.....	(45)
3.3.3.1	提供菜单控制规则.....	(46)
3.3.3.2	建立菜单控制规则的模型.....	(47)
3.3.4	显示弹出式子菜单.....	(48)
3.3.5	从弹出式子菜单中生成一个事件.....	(48)
3.4	处理菜单控制规则.....	(48)
3.4.1	预处理:第一步	(49)
3.4.2	后处理:两次扫描操作	(51)
3.4.2.1	函数 MenuEvent	(54)
3.4.2.2	函数 CheckPick	(55)
3.4.2.3	函数 IfAnyPost	(55)
3.4.3	设置菜单选项标志和消息.....	(57)
3.4.3.1	过程 SetPopup	(57)
3.4.3.2	过程 SavePop	(60)
3.4.3.3	过程 SetFMPad1	(61)
3.4.3.4	使用函数控制菜单选项.....	(65)
3.4.3.5	过程 FMPad1Msg	(66)
3.4.3.6	过程 FMPad1Pop	(67)
3.4.3.7	菜单控制规则的处理流程.....	(70)
3.5	ON SELECTION MENU 和 ON PAD	(70)

3.6 使用菜单按钮.....	(71)
3.6.1 使菜单按钮可用.....	(72)
3.6.2 产生菜单事件.....	(73)
3.6.2.1 控制菜单按钮.....	(74)
3.6.2.2 设置菜单按钮标志.....	(74)
3.6.2.3 重新设置菜单按钮.....	(76)
3.6.3 在窗口内加上按钮.....	(76)
3.7 使用运行菜单.....	(78)
 第4章 窗口管理	(82)
4.1 激 活 窗 口.....	(83)
4.1.1 离开窗口.....	(83)
4.1.2 返回窗口.....	(83)
4.2 识别四种类型的信息.....	(86)
4.2.1 环境信息.....	(86)
4.2.2 当前窗口信息.....	(88)
4.2.3 系统信息.....	(89)
4.2.4 当前记录信息.....	(89)
4.2.4.1 编辑处理的三种方式.....	(89)
4.2.4.2 影象文件.....	(90)
4.2.5 运行状态.....	(91)
4.3 使用系统管理程序.....	(92)
4.3.1 指定事件的操作.....	(93)
4.3.2 通用窗口管理操作.....	(96)
4.3.3 把窗口看作对象.....	(97)
4.4 完 成 循 环.....	(97)
4.5 多窗口程序设计主要问题总结.....	(99)
4.5.1 事件管理问题(第2章).....	(99)
4.5.2 菜单管理问题(第3章).....	(99)
4.5.3 窗口管理问题(第4章)	(100)
4.5.3.1 当前环境	(100)
4.5.3.2 当前窗口	(101)
4.5.3.3 整个系统	(101)
4.5.3.4 事件信息	(101)
4.5.3.5 当前记录	(101)

第二部分 构造应用程序

第5章 人 门	(102)
----------------------	--------------

5.1 演示应用程序	(102)
5.1.1 家族关系记录	(103)
5.1.1.1 使信息可存取	(103)
5.1.1.2 使信息可用	(103)
5.1.2 空文件或空记录集处理	(104)
5.2 文 件	(104)
5.2.1 用户数据文件	(104)
5.2.1.1 Companies 文件	(105)
5.2.1.2 Contacts 文件	(105)
5.2.1.3 Log(Actions)文件	(106)
5.2.1.4 Events 文件	(106)
5.2.1.5 Events Detail 文件	(106)
5.2.2 系统管理文件	(107)
5.2.2.1 记录系统运行状态:系统文件	(107)
5.2.2.2 记录用户信息:Personnel 文件	(107)
5.2.2.3 跟踪记录锁和相关记录:Pseudo-Lock 文件	(108)
5.2.2.4 执行查询:Lookup 文件	(109)
5.2.2.5 提供帮助:Help 文件	(109)
5.2.2.6 保存处理中的状态:File Stack(文件栈)	(109)
5.2.3 影象文件	(110)
5.3 索引标记(TAG)	(110)
5.4 菜 单	(111)
5.5 变量与数组	(112)
5.6 事 件	(112)
第6章 事件管理	(114)
6.1 使用事件循环:过程 EventLoop	(114)
6.2 使用通用设置例程:过程 EnvSetup	(115)
6.3 使用通用 BROWSE 管理程序:过程 GoBrowse	(116)
6.4 捕获记录更换事件:函数 IfRecChng	(119)
6.5 使用通用 READ 管理程序:过程 GoEdit	(120)
6.6 事件捕获与确定:函数 EventTrap	(124)
6.6.1 捕获窗口关闭事件:函数 OnWClose	(126)
6.6.2 捕获菜单选择事件:函数 OnWMenu	(126)
6.7 捕获按钮选择事件:函数 OnWButton	(126)
6.7.1 捕获热键事件:函数 OnWHotkey	(127)
6.7.2 捕获窗口开关事件:函数 OnWSelect	(128)
6.7.3 捕获超时事件:函数 OnWTime	(128)