

ArEM

Visual FoxPro 3.0 中文版系列丛书之一



从 FoxPro 2.X 到 Visual FoxPro 3.0 中文版

孟刚 编著

马晓红 韩冬晖 王兆其 审校



WQ

清华大学出版社

阿特曼电脑之友丛书

Visual FoxPro 3.0 中文版系列丛书之一

从 FoxPro 2.X 到 Visual FoxPro 3.0 中文版

孟 刚 编著

马晓红 韩冬晖 王兆其 袁国

清华大学出版社

(京)新登字 158 号

JS170/53
内容简介

本书分三大部分，全面、系统、完整地介绍 Visual FoxPro 3.0 中文版。首先介绍了 Windows 编程与面向对象技术的基本概念。然后在涵盖所有 Visual FoxPro 3.0 有别于 FoxPro 2.X 的知识点的前提下，使读者对 Visual FoxPro 3.0 有较全面、深层的理解，对 Visual FoxPro 3.0 中应用程序的开发过程有一个整体的认识，帮助读者把已有的经验移植到 Visual FoxPro 3.0 中。本书语言简捷，通俗易懂。

本书可作为高校计算机专业的学生、计算机程序设计人员以及从事计算机研究、开发工作人员的参考书。

版权所有，翻印必究

本书封面贴有清华大学出版社激光防伪标志，无标志者不得销售。

书名：从 FoxPro 2.X 到 Visual FoxPro 3.0 中文版

作者：孟刚

出版者：清华大学出版社(北京清华大学校内，邮编：100084)

印刷者：北京市丰台区丰华印刷厂印刷

发行者：新华书店总店科技发行所

开本：787×1092 1/16 印张：22.25 字数：509.7 千字

版次：1996年12月第1版 1997年7月第3次印刷

书号：ISBN 7-302-02117-1/TP·995

印数：10001~15000

定价：36.00 元

丛书序

FoxPro 是近年来出现的一种重要的数据库产品。一经问世，它就引起了广大的数据库用户和开发人员的极大兴趣，同时获得使用者的一致好评。从最初的 Fox 产品发展到现在的集界面设计、程序开发和数据操作于一身的数据库管理系统，FoxPro 因为友好的用户界面、方便的程序开发工具以及强大的数据处理功能而倍受人们的青睐。

Microsoft Visual FoxPro 3.0 中文版作为最新版本的 Xbase 系列数据库管理系统，在功能和特色上均有较大的改进。具体表现在如下几方面：

1. 改进的用户界面。经过大量的改进，用户界面更易发挥 FoxPro 的功能。
2. 数据库性能的增强。数据库不仅具有客户机/服务器能力，而且能支持 NULL 值，并提供事务处理。
3. 语言的改进。在语言方面，相应增加了一些内容来支持新的功能，其中包括面向对象的程序设计，同时支持 OLE 自动化。
4. 新的数据库容器。Visual FoxPro 独具特色的数据库容器，为交互式用户和应用程序开发者提供了集中的数据管理功能。
5. 汉字处理功能。在 Visual FoxPro 3.0 中文版中，文本性质的各类名称都可以使用中文来命名。

针对该版的特点，凭我们在汉化过程中对 Visual FoxPro 的理解以及对 FoxPro 2.X 程序员的准确定位，我们编写了一系列的丛书，对 Visual FoxPro 3.0 中文版进行了系统而详尽的介绍：

- 《从 FoxPro2.X 到 Visual FoxPro 3.0 中文版》
- 《Visual FoxPro 3.0 中文版程序设计指南》
- 《Visual FoxPro 3.0 中文版命令手册》
- 《Visual FoxPro 3.0 中文版函数手册》
- 《Visual FoxPro 3.0 中文版类和对象手册》

丛书的主要读者对象是微机数据库系统的开发者和一般用户，相关领域的工程技术人员也可将其作为参考书使用。

由于我们水平有限，加上时间匆促，书中难免有不少错误和不足，欢迎广大的读者给予批评和指正。

编者

1996年1月

前言

时势造英雄

时势也造就了 Visual FoxPro 3.0，这些时势有软件开发思想和计算模式的进步，有计算机软硬件的发展，有不同数据库管理系统之间的竞争。

面向对象方法的基本概念经历了 10 年时间走向了成熟，对这一方法的使用也达到了高潮。Visual FoxPro 3.0 完成了从面向字符式数据库到面向对象风格数据库的转换。是不是可以说 Visual FoxPro 3.0 赶上潮流了呢？不可以，因为这种转换决不是为了赶时髦。今天我们建造的数据库系统已不同于 10 年或 20 年以前的了。在任何方面，他们都变得更庞大更复杂更加不稳定和更易变化。并且用户对界面的要求越来越高，当今系统中高达 75 % 的代码是与界面有关的。而实践证明，面向对象方法是同时解决上述两个问题的一种更自然的方法，这才是 Visual FoxPro 3.0 转向面向对象风格的真正原因。

人们在争先恐后连接计算机网络的同时，也逐渐接受了网络环境中最佳的计算模式—客户/服务器结构。这促使 Visual FoxPro 提供开发客户/服务器应用程序的功能。您可以把 Visual FoxPro 3.0 作为前端并可以通过 SQL 传递功能使用本地机的语法来访问服务器，指示服务器完成所需操作。为保证多用户环境中对数据访问的安全性，Visual FoxPro 3.0 中增加了一个数据字典，提供了从字段到表级到事务级的安全性控制。

时不我待

您已经具有了 FoxPro 2.X 程序的开发经验，转到 Visual FoxPro 3.0 后怎样抓住最主要的内容而不必在已了解的知识上浪费时间呢？答案就是找一本过渡性的书，它应该能够把 Visual FoxPro 3.0 中新增的功能讲解清楚，并能引导您把 FoxPro 2.X 的开发经验移植到 Visual FoxPro 3.0 中。

上者伐谋，中者伐兵，下者伐城

似乎找出所有 Visual FoxPro 3.0 有别于 FoxPro 2.X 的地方并把它们罗列出来，这本书的目的就达到了，但我们的要求绝没有这么低。

一个人接受新知识的快慢，可以从他的注意点在哪儿看出来。正确的方法是自顶向下，先对新事物形成一个整体的认识，搞清楚自己已经知道什么，还需要知道什么，然后把注意力集中在所需的细节上。不过见了太多的人一头扎在一个个独立的知识点上，不过问它们之间的联系，不知道为什么学习它们，这样即使搞懂了 Visual FoxPro 3.0 与 FoxPro 2.X 之间的一千零九十八个

差别，拿到一个项目后还是无从下手。所以写这本书时我还有一个更深层的目的，那就是能使读者把已有的知识与 Visual FoxPro 3.0 新增的知识组织起来，对 Visual FoxPro 中应用程序的开发过程有一个整体的认识。

基于这样的考虑，本书把系统性、完整性放在第一位，在涵盖所有 Visual FoxPro 3.0 有别于 FoxPro 2.X 的知识点的前提下，力争使读者对 Visual FoxPro 3.0 有一个全面的理解。达到这个目的的“谋”主要有以下三点，分别对应着本书的三个部分：

1. 在第一部分解决 Visual FoxPro 3.0 相对 FoxPro 2.X 的两个大块的改进：全面的 Windows 环境以及面向对象的风格。介绍两者时都是首先从广义上讲清概念，然后再具体到 Visual FoxPro 3.0，以使读者高屋建瓴，对 Visual FoxPro 有深层的理解。
2. 第二部分中，按创建 Visual FoxPro 3.0 应用程序的过程把所有其他小的不同点组织在一起。这一部分特别注意了结构的完整性，可以保证：即使是第一次接触 FoxPro 的读者，只要有一定领悟能力，也能掌握 Visual FoxPro 3.0。
3. 对于 FoxPro 的老用户，第三部分使您有一个进阶的机会。您可以学习如何在 Visual FoxPro 与其他应用程序之间以及不同用户之间交换信息。

Every Dog Has It's Day

这本书能有它的今天，首先要归功于 ArtM 公司，正是她提供的宽松环境使得本书得以顺利完成。感谢 ArtM 公司中所有的 Art Man（当然还有所有的 Art Woman），他们或提供资料、或帮助释疑，或辅以排版、或助之插图，他们的支持是这本书得以出版最直接的因素。特别感谢孙立林、廖健飞、杨颖波先生以及郑丽莎小姐，需要时他们总是提供最无私的帮助。

感谢 ArtM 公司总经理韩冬晖先生以及微软（中国）有限公司的马晓红小姐。还要感谢清华大学出版社的徐培忠及陈克强老师，他们不仅给本书以出版的机会，更在内容及编排方面提出建设性的意见。最后不能不提及的是已在美国的李宙华先生，感谢他的相惜之情。

孟刚

1996 年 4 月

目 录

第一部分 排除障碍,进入 Visual FoxPro 3.0 世界.....1

第 1 章 从 DOS 进入 Windows2

1.1 WINDOWS 编程环境.....	3
1.1.1 图形用户界面 GUI.....	3
1.1.2 图形设备接口 GDI.....	4
1.1.3 多任务处理.....	4
1.1.4 信息交换(DLL,DDE,OLE).....	5
1.1.5 内存管理.....	6
1.1.6 True Type 字型技术.....	6
1.1.7 网络支持.....	7
1.1.8 多媒体支持.....	7
1.1.9 开放数据库连接 ODBC.....	7
1.2 WINDOWS 编程机制.....	8
1.2.1 控制及对象的概念.....	8
1.2.2 事件驱动的程序设计.....	9
1.2.3 消息循环和处理机制.....	10
1.3 WINDOWS 软件开发工具.....	14
1.3.1 元件开发工具(Component Builder).....	16
1.3.2 解决方案开发工具(Solution Builder).....	16
1.3.3 决策支持及查询工具(Decision Support and Query Tools).....	17

第 2 章 Visual FoxPro 3.0 特性 19

2.1 快速创建应用程序的能力.....	19
2.1.1 新添的向导和生成器.....	19
2.1.2 获得快捷键的工具栏.....	20
2.1.3 创建应用程序元件的设计器.....	21
2.1.4 不用编程而创建应用程序界面.....	21
2.1.5 使用项目管理器管理工作.....	22
2.2 更强大的开发能力.....	22
2.2.1 充分利用面向对象的程序设计.....	22
2.2.2 灵活地处理事件.....	23
2.2.3 优化您的系统.....	24
2.2.4 优化检索过程的 Rushmore 技术.....	24

2.3 开发客户/服务器程序	24
2.3.1 定义规则的数据字典	24
2.3.2 查看远程或异构数据	25
2.3.3 用事务控制共享访问	25
2.3.4 实现客户/服务器应用程序	26
2.4 与其他应用程序相互作用	26
2.4.1 与其他应用程序共享数据	26
2.4.2 导入和导出数据	27
2.4.3 使用 OLE 自动化控制其他应用程序	27
2.5 VISUAL FOXPRO 3.0 软件开发思想	27
2.5.1 从编码转入分析设计	28
2.5.2 快速原形的思想	29

第 3 章 从面向过程到面向对象..... 31

3.1 对象	31
3.1.1 对象的概念	31
3.1.2 对象的特性——属性	34
3.1.3 对象的方法及相关事件	35
3.2 理解类	36
3.2.1 类	36
3.2.2 子类 subclass	37
3.2.3 继承 inheritance	40
3.2.4 多态性 polymorphism	41

第 4 章 Visual FoxPro 中的对象及类 44

4.1 VISUAL FOXPRO 中的类层次	44
4.1.1 Visual FoxPro 基类	44
4.1.2 容器与非容器	46
4.2 操纵对象	47
4.2.1 通过容器层次引用对象	48
4.2.2 设置属性值	49
4.2.3 读取属性值	50
4.2.4 调用方法	51
4.2.5 响应事件	51
4.3 以编程方式定义类	52
4.3.1 指定类的名称及父类	52
4.3.2 保护类成员	53
4.3.3 向容器类中添加对象	54

4.3.4 指定方法代码和事件代码	55
4.3.5 实例分析: 创建表定位按钮集	57
4.4 处理对象及数据	62
4.4.1 创建对象引用	63
4.4.2 创建成员数组	64
4.4.3 使用对象存储数据	65

第 5 章 Visual FoxPro 的事件模型 68

5.1 基本概念	68
5.1.1 模式及无模式操作	68
5.1.2 与 FoxPro 2.X 比较	68
5.2 VISUAL FOXPRO 中的事件	69
5.3 在容器和类层次中事件的处理	70
5.3.1 容器和对象事件	70
5.3.2 类和控制事件	72
5.4 实例分析: 事件发生顺序	72
5.5 编写事件代码	75

第 6 章 进入 Visual FoxPro 3.0 世界 77

6.1 使用 VISUAL FOXPRO 的工具	77
6.1.1 使用项目管理器	77
6.1.2 使用 Visual FoxPro 设计器	82
6.1.3 使用工具栏	82
6.1.4 使用向导	83
6.2 VISUAL FOXPRO 3.0 导游	84
6.2.1 主旨	84
6.2.2 所需设置	84
6.2.3 操作步骤	84

第二部分 在 Visual FoxPro 3.0 中开发应用程序 93

第 7 章 Visual FoxPro 应用程序的开发过程 94

7.1 规划应用程序	94
7.1.1 应用程序创建过程概述	95
7.1.2 用面向对象的方法进行分析及设计	96
7.2 创建数据库	98
7.3 提供访问信息的手段	98

7.4 设计并创建类	99
7.5 提供访问应用程序功能的手段	99
7.6 测试和调试	99

第 8 章 设计数据库 100

8.1 数据库包含文件 .DBC 包含什么	100
8.2 处理表	103
8.2.1 数据库表与自由表的区分	103
8.2.2 字段类型的增加	106
8.2.3 索引方式的完善	109
8.2.4 数据字典功能的应用	112
8.2.4.3 使用触发器	120
8.3 使用视图	124
8.3.1 创建本地视图	125
8.3.2 创建远程视图	128
8.3.3 限制视图的作用范围	131
8.3.4 使用视图	133
8.3.5 在视图中更新数据	137
8.3.6 集成本地视图和远程视图	144
8.3.7 优化视图性能	145
8.3.8 使用视图中的数据字典功能	147
8.4 使用查询及报表	149
8.4.1 使用查询	149
8.4.2 使用报表	152
8.4.3 数据之间的关系	156

第 9 章 设计界面 157

9.1 从屏幕到表单	157
9.1.1 设置数据环境	157
9.1.2 向表单添加控制	159
9.1.3 向表单或表单集添加属性和方法	162
9.1.4 定制表单上的控制	164
9.1.5 定制表单的行为	167
9.1.6 在运行时设置属性或调用方法	175
9.1.7 处理表单数据	178
9.1.8 实例分析	179
9.2 使用控制	183
9.2.1 控制与数据的关系	183

9.2.2 选择适当的控制	184
9.3 设计自定义工具栏	213
9.3.1 定义工具栏类	213
9.3.2 在自定义工具栏类中添加对象	215
9.3.3 在表单集中添加自定义工具栏	215
9.3.4 实例分析	216
9.4 设计菜单	218
9.4.1 重新理解菜单系统	219
9.4.2 设计菜单	220
9.4.3 给菜单或菜单项指定任务	221
9.4.4 协调使用菜单和工具栏	221

第 10 章 设计并创建类..... 222

10.1 根据任务定义类	222
10.1.1 决定什么情况下创建类	222
10.1.2 决定要创建的类的类型	223
10.2 创建类	225
10.2.1 使用类库文件	225
10.2.2 创建新类	226
10.2.3 调用父类方法	235
10.3 将类添加到表单、表单集和工具栏中	237
10.4 修改类定义	238
10.5 创建类定义的子类	239
10.6 以编程方式创建类的对象	239

第 11 章 编写应用程序代码..... 240

11.1 什么地方需要代码	240
11.1.1 什么地方需要代码?	240
11.1.2 什么时候使用面向对象的编程?	244
11.2 VISUAL FOXPRO 语言的改进	245
11.2.1 数据类型的增加	245
11.2.2 变量的作用域	246
11.2.3 使用类及对象	252
11.2.4 处理 NULL 值	252
11.3 VISUAL FOXPRO 3.0 与 FOXPRO 2.6 术语的区别	255
11.4 命名规则	255
11.4.1 变量命名约定	256
11.4.2 对象命名约定	257

11.4.3 表字段命名约定	258
11.4.4 常量命名约定	258
11.4.5 窗口命名约定	258
第 12 章 编译应用程序.....	259
12.1 构造应用程序框架	259
12.1.1 设置起始点	260
12.1.2 设置应用程序环境	261
12.1.3 显示初始界面	262
12.1.4 控制事件循环	262
12.1.5 恢复原始环境	263
12.2 将文件加入到项目中	263
12.3 连编项目并显示错误	264
12.4 连编应用程序	265
引用可修改的文件	265
12.5 运行应用程序	266
12.6 测试组件	266
第三部分 扩展应用程序.....	268
第 13 章 与其他应用程序相互作用	269
13.1 导入和导出数据	269
13.1.1 理解导入和导出	269
13.1.2 导入数据	271
13.1.3 追加数据	273
13.1.4 导出数据	276
13.2 链接和嵌入数据	281
13.2.1 向应用程序中添加 OLE 对象	282
13.2.2 用 OLE 自动化技术创建对象	286
13.2.3 操作 OLE 对象	290
第 14 章 多用户环境下的访问控制	296
14.1 数据访问的控制	296
14.1.1 访问数据	296
14.1.2 锁定数据	298
14.1.3 使用数据工作期	302
14.2 数据访问的缓冲	304

14.2.1 缓冲编辑结果	304
14.2.2 执行更新	309
14.2.3 管理冲突	310
14.2.4 检测并解决冲突	312
14.3 用事务管理更新操作	314
14.3.1 包装代码段	314
14.3.2 控制事务处理的命令	314
14.3.3 保护远程更新	319
14.3.4 性能管理	321
附录 A 转换 FoxPro 2.6 文件	322
附录 B Visual FoxPro 文件的表结构	332
附录 C Visual FoxPro 3.0 中的控制	341

第一部分

排除障碍，进入 Visual FoxPro 3.0 世界

对 FoxPro 2.X 编程人员来说，熟悉 Windows 编程环境，面向对象的程序设计和事件循环的编程机制是掌握 Visual FoxPro 3.0 编程的三个障碍。本部分内容在解决这三大障碍的基础上力图使读者对 Visual FoxPro 有一个整体的认识。

第 1 章 从 DOS 进入 Windows

介绍 Windows 编程环境，比如图形用户界面、多窗口多任务等等。重点介绍 Windows 的事件驱动原理。

第 2 章 Visual FoxPro 3.0 特性

介绍 Visual FoxPro 3.0 新增的功能及特性，使读者搞清楚自己知道什么以及不知道什么。

第 3 章 从面向过程到面向对象

从广义上介绍面向对象方法中的概念及思想，Visual FoxPro 不仅是一个面向对象的编程语言，更是面向对象分析及面向对象设计的工具。

第 4 章 Visual FoxPro 中的对象及类

具体介绍 Visual FoxPro 所支持的对象及类以及它们的使用方法。

第 5 章 Visual FoxPro 的事件模型

事件模型介绍了用户如何激发事件以及 Visual FoxPro 什么时候、如何对事件做出响应。

第 6 章 进入 Visual FoxPro 3.0 世界

在介绍 Visual FoxPro 常用工具之后，通过一系列简短的步骤带领读者浏览 Visual FoxPro 3.0 的界面使用及主要特性。

第 1 章 从 DOS 进入 Windows

读到这本书的时候，您也许早已成为 6000 万 Windows 用户中的一员。比起 DOS “漆黑” 的字符环境，Windows 提供的图形化操作平台着实吸引着每一个人。于是我们纷纷抛弃 DOS，在机器中安装 Windows，并在几天之内将操纵鼠标的技巧练得炉火纯青。但这就是如同本章题目所说的，从 DOS 进入 Windows 了吗？

仅仅使用 Windows 的文件管理器、画笔等应用程序，不是真正意义上的“进入 Windows”，因为这仅仅是使用 Windows 的管理或附加功能，就像在 DOS 中只是使用 DIR 或 FORMAT 等命令；在 Windows 中编程，比如双击 QBASIC 图标进入 BASIC 编一个求阶乘的程序，也不算进入 Windows。实质上，一旦进入 BASIC，系统就被 DOS 接管；什么时候您开始希望拿 Windows 做些什么，比如编制一个带有窗口、菜单等界面的 Windows 程序，就说明您已准备从一个 Windows 的用户发展成 Windows 程序员了。

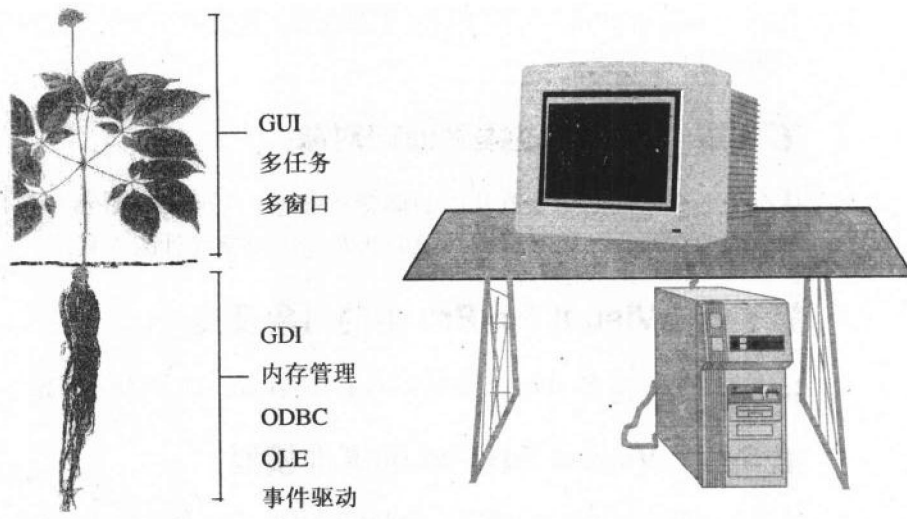


图1-1 Windows是一株人参

Windows 是一株人参（见图 1-1）：对最终用户来讲，它呈现的只是地面之上漂亮的花朵。只有 Windows 程序员才能得到地面之下最有价值的东西，才能充分发挥它的能力。从用户的角度讲，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，并能够在各个应用程序之间快速切换。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，事件驱动的编程方式，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，同时对 Windows 本身进行增值。

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 GUI 以及与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。微软及第三方厂商纷纷开发 Windows 软件开发工具。VisualFoxPro3.0 就是其中用来开发数据库应用程序的软件。

本章讨论以下几个方面的内容:

- Windows 编程环境
- Windows 编程机制
- Windows 软件开发工具
- Visual FoxPro 3.0 软件开发思想

1.1 Windows 编程环境

环境指的是 Windows 本身或 Windows 应用程序提供给用户的交互方式,比如菜单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。程序员编制 Windows 应用程序时所用的软件开发工具以及 Windows 本身提供的资源构成了 Windows 的编程环境。本节介绍 Windows 提供的编程环境及资源。

1.1.1 图形用户界面 GUI

Windows 的图形用户界面 (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 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。

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，所以可使同一函数在点阵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。Windows 为此提供了一种图形设备接口（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，如图 1-2 所示。

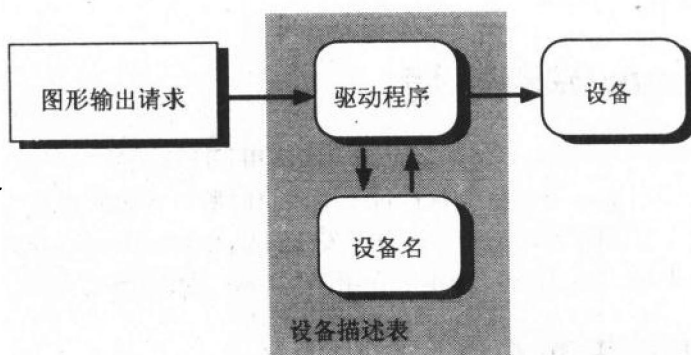


图1-2 图形输出请求的处理

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。

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 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