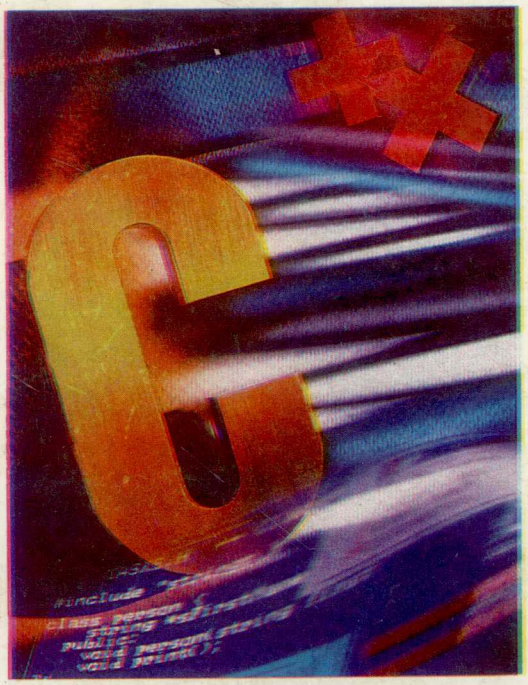
BORLAND C++3.0/3.1
界面设计技术与范例

冯矢勇　主编
倩玉枫　主校



Borland C++

- 界面设计导论
- Turbo Vision 的要点
- Turbo Vision 的应用
- Turbo Vision 的应用设计
- Resource Workshop 的应用功能
- Object Windows 的要点
- Object Windows 的应用
- Object Windows 应用设计

学苑出版社

BorlandC++3.0/3.1

界面设计技术与范例

冯矢勇 编著
倩玉枫 审校

学苑出版社
1993.

（京）新登字151号

内 容 提 要

C++是面向对象程序设计语言中最成熟和广泛于普及的一种。C++教程共分三册（另两册是Borland C++3.0基础教程；C++3.0应用中级教程）。本书是第三册，其主要内容是DOS下和Windows3.0／3.1下的界面设计。设计方法不是停留在C／C++语言级，而是充分利用Borland C++3.0／3.1中TurboVision及ObjectWindows两大工具包，因而使软件具有精练性、一致性和高度重用性。本书除了阐明设计的要点及上述工具的主要应用方法外，还叙述实际应用场合的设计要点。附有众多的实例程序。本书是计算机软件应用及开发人员、研究生及大学生等的重要参考资料，也可作为专业人员的教材。

计算机程序设计语言系列丛书

BorlandC++3.0／3.1界面设计技术与范例

编　　著：冯矢勇
审　　校：倩玉枫
责任编辑：徐建军
出版发行：学苑出版社　　邮政编码：100032
社　　址：北京市西城区成方街33号
印　　刷：常熟市教育印刷二厂
开　　本：787×1092　1/16
印　　张：33.38　　字数：774千字
印　　数：1—5000册
版　　次：1993年11月北京第1版第1次
ISBN7—5077—0807—1/TP·18
本册定价：49.00元（含盘）

前　言

“软件产业的革命”，这是国内外计算机专家和软件专家对面向对象技术的看法。而C++语言是面向对象程序设计语言（OOP）中最成熟、最适宜于普及应用的工具，它是一颗正在上升的耀眼的高级语言新星。

首先，C++受到C语言程序员和软件开发人员的欢迎。今后将要受到各个领域（如管理、经济、工程技术、科学研究等）中越来越多专家的接受和喜爱。

C语言程序员之所以欢迎C++，是因为只要学习几天就可利用C++“小规模的优点”，在软件开发中收到立竿见影的效果。

软件开发人员之所以欢迎C++，是因为他们发现C++就是梦寐已求的开发大型复杂软件的工具。

C++程序设计的方法和传统的方法有根本不同：

OOP软件设计的关键在于上层设计，而不是它的实现和细节。所以这种特点会使越来越多的非软件领域的专家所欢迎和接受。从某种角度上说，他们对软件的上层设计能提出比程序员更合理的方案。

可以预见，C++不但将普及到程序员，并且将普及到所有对计算机技术有兴趣的人员。这样，C++可成为软件人员和其他领域工作者共同相通的语言，这个意义是很巨大的。

本教程共分三册出版，第一册是C++的入门基础教程，第二册是C++应用方面的中级教程，第三册是C++专题应用技术的高级教程：界面设计。

本书是第三册，是一本C++专题应用技术——界面的设计技术。国内已经问世的有以C（或C++）语言开发界面的文献，但是以语言级为基础的设计，对程序员来说，既是一个沉重的负担，又是一种个体艺术——不适用于软件生产。本书阐述如何充分利用Borland C++3.0/3.1中的Turbo Vision及ObjectWindows等工具包设计DOS的界面以及Windows的界面，并使这样的软件设计具有高效率、精练性、一致性及高度重用性。

本书在编写出版过程中，由裘迅、黄敏宇、祝勇及卢琳琪等同志协助整理和校对了部分文稿，在此向他们表示感谢。

编者衷心感谢读者的批评及建议，并欢迎进行交流。来函请寄：邮编100080，北京8721信箱资料部。

编者

1993年国庆节前于苏州

引　言

本书是DOS界面及Windows界面的系统设计方法专著；更具体的是利用 BorlandC++3.0／3.1（对大部分问题也可用2.0版本）的Turbo Vision,ObjectWindows以及 Resource Workshop 工具，设计用户界面。利用工具设计出来的界面具有显著优越性：开发时间可缩短好几倍（与C语言开发比较）、界面保持一致（美观、清晰及通用）以及可继承可重用。著者曾用Turbo Vision开发一个工程的DOS界面只用了两天；而通常用C语言需要一、二十天。但是，要掌握工具，需要投入较多的精力，因为它们各有一个庞大的类继承机制。例如，美国出版了一本“用Borland C++3开发Windows应用程序”，专门阐述 Object Windows，共计1362页，几乎是一本“巨著”。可见界面开发，如果要用工具，是相当方便，但是，要掌握它们，却并非易事。

本书从知识角度可分为三大部分：

本书第一部分是属于背景知识及常用参考，分别列于第一章的用户界面设计导论，第五章的Resource Workshop的功能以及最后的几个附录。

本书第二部分是论述利用Turbo Vision设计DOS应用界面的方法。第二章是阐明 Turbo Vision工具本身的要点：特点、类及功能。第三章是由浅入深地说明各种界面元素的应用技术。第四章论述应用程序如何进行界面设计。

本书第三部分是论述利用ObjectWindows设计Windows应用界面的方法。第六章是阐明ObjectWindows工具本身的要点：特点、类及功能。第七章是深入浅出地说明各种界面元素的应用技术。第八章论述应用程序的界面设计方法。

第二部分和第三部分几乎各自独立的。读者可按需选用。

本书提供了很多格式几乎一致的实例程序。为便于读者利用，可为读者提供它们的源程序（附盘1张）。本书的例子程序是针对Brorland C++3.1及 Wincows3.1的，一般可在 Borland C++3.0及Windows3.0下使用，如有差别，将在文中说明。

目　录

第一章　界面设计导论

1.1 Windows的影响 …… (2)
1.1.1 Windows应用特点 …… (2)
1.1.2 与DOS完全不同的四个概念 …… (3)
1.1.3 Windows应用程序设计的注意事项 …… (7)
1.2 Borland C++的界面工具 …… (7)
1.3 继承关系表示法 …… (11)
1.3.1 图形表示法 …… (11)
1.3.2 缩排表示法 …… (11)
1.3.3 缩排方式的优点 …… (14)
1.4 不要再发明轮子 …… (14)
1.4.1 轮子已经发明 …… (14)
1.4.2 Borland C++界面工具远优于C库函数 …… (15)
1.5 一些约定 …… (15)
1.5.1 例子程序中的标记 …… (15)
1.5.2 用户标识符 …… (16)
1.5.3 Turbo Vision命名的约定 …… (16)
1.5.4 Object Windows命名的约定 …… (17)
1.6 掌握Borland界面工具的方法 …… (18)

第二章　Turbo Vision的要点

2.1 TV的三个主要特征 …… (19)
2.1.1 事件 …… (19)
2.1.2 视口 …… (19)
2.1.3 哑对象 …… (19)
2.2 类的继承体系 …… (20)
2.2.1 继承体系表 …… (20)
2.2.2 TV类的种类 …… (20)
2.2.3 TV类的成员 …… (21)
2.2.4 类的实例和派生 …… (23)
2.2.5 三个原始基类 …… (23)
2.2.6 视口 (TViews) 类 …… (25)

2.3 TV类的头文件及TV.H的特点 …… (29)
2.3.1 TV类的INCLUDE子目录下的文件 …… (29)
2.3.2 TV.H的内容 …… (30)
2.3.3 使用TV头文件的特点 …… (31)
2.4 菜单和状态行的设计 …… (31)
2.4.1 设计要求 …… (31)
2.4.2 设计要点 …… (32)
2.4.3 实例程序TVFCS—01.CPP代码 …… (35)
2.4.4 项目文件.PRJ的设计 …… (38)
2.4.5 编译注意 …… (38)
2.4.6 运行 …… (38)
2.5 创建窗口及文件读入的设计 …… (38)
2.5.1 设计要求 …… (38)
2.5.2 设计要点 …… (38)
2.5.3 实例程序TVFCS—02.CPP代码 …… (42)
2.5.4 补充说明 …… (47)
2.6 滚动条及空白对话框的设计 …… (49)
2.6.1 设计要求 …… (49)
2.6.2 设计要点 …… (50)
2.6.3 实例程序TVFCS—03.CPP …… (51)
2.7 对话框内容 …… (59)
2.7.1 设计要求 …… (59)
2.7.2 设计要点 …… (59)
2.7.3 实例程序TVFCS—04.CPP …… (60)
2.7.4 补充说明 …… (68)

第三章 Turbo Vision的应用

3.1 视口的设计 …… (74)
3.1.1 屏幕交给了Turbo Vision …… (74)
3.1.2 简单视口 …… (74)
3.1.3 复杂视口（组视口） …… (75)
3.1.4 子视口的Z序 …… (76)
3.1.5 平台的Z序 …… (77)
3.1.6 视口之间的关系 …… (77)
3.1.7 属主视口和子视口 …… (78)
3.1.8 选中视口 …… (78)
3.1.9 模式视口 …… (79)
3.1.10 修改默认性能 …… (79)
3.1.11 视口的颜色设置 …… (81)

3.2 设计安全的程序 …… (83)
3.2.1 安全池 …… (84)
3.2.2 应该使用destroy ( ) …… (85)
3.2.3 处理大用户 …… (85)
3.2.4 非存储错误问题 …… (85)
3.3 设计事件驱动程序 …… (86)
3.3.1 事件的实质 …… (86)
3.3.2 事件检测例程 …… (88)
3.3.3 事件的处理 …… (91)
3.3.4 修改事件的机理 …… (92)
3.3.5 命令 (Commands) …… (97)
3.3.6 视口内的通讯 …… (98)
3.4 收集类和可流类的应用 …… (101)
3.4.1 两种类系的功能 …… (101)
3.4.2 类的继承体系 …… (102)
3.4.3 主要类的结构 …… (102)
3.4.4 应用实例 …… (109)
3.5 调试就是设置断点 …… (126)
3.5.1 系统挂起问题 …… (126)
3.5.2 检查指针 …… (126)
3.5.3 无法设置断点的地方 …… (127)
3.5.4 事件的失踪 …… (127)
习题 …… (128)
参考答案 …… (133)

第四章 Turbo Vision的应用设计

4.1 几点有益的建议 …… (136)
4.1.1 继承是法宝 …… (136)
4.1.2 充分利用IDE …… (136)
4.1.3 充分利用Turbo Vision中的例子程序 …… (137)
4.1.4 安排好设计中的各个细节 …… (138)
4.2 界面的基本模块设计 …… (139)
4.2.1 头文件的设计 …… (140)
4.2.2 定义模块代码 …… (143)
4.2.3 应用模块 …… (150)
4.2.4 另一种设计方式 …… (151)
4.3 工程应用界面的设计 …… (151)
4.3.1 头文件设计 …… (151)
4.3.2 系统模块程序设计 …… (154)
4.3.3 应用模块程序设计 …… (165)

4.3.4 实际任务的设计 ……（168）
4.4 编辑应用界面设计 ……（180）
4.4.1 增加的功能 ……（181）
4.4.2 头文件设计 ……（181）
4.4.3 编辑模块程序设计 ……（182）
4.4.4 简单应用程序设计 ……（189）
4.4.5 扩展应用程序的设计 ……（190）

第五章 Resource Workshop的应用功能

5.1 功能 ……（200）
5.2 启动 ……（200）
5.2.1 硬件要求 ……（200）
5.2.2 软件要求 ……（200）
5.2.3 RW包含的文件 ……（201）
5.2.4 RW的安装 ……（201）
5.2.5 启动RW ……（201）
5.2.6 RW的菜单条 ……（201）
5.3 资源及资源文件 ……（202）
5.3.1 何为资源 ……（202）
5.3.2 资源类型 ……（202）
5.3.3 资源文件 ……（205）
5.3.4 管理项目文件 ……（205）
5.4 文本编辑器 ……（205）
5.5 图像编辑器（Paint） ……（206）
5.5.1 启动Paint编辑器 ……（206）
5.5.2 功能 ……（206）
5.5.3 颜色 ……（206）
5.5.4 工具箱 ……（207）

第六章 ObjectWindows的要点

6.1 一个原始的框架程序（OWFCS-00.CPP） ……（214）
6.1.1 程序功能 ……（214）
6.1.2 程序清单与设计 ……（215）
6.1.3 调试问题 ……（218）
6.1.4 程序说明 ……（219）
6.2 ObjectWindows类的继承机制 ……（221）
6.2.1 类表及主要类 ……（221）
6.2.2 Object类（OBJECT.H） ……（221）
6.2.3 TModule类（MODULE.H） ……（222）
6.2.4 TApplication类（APPLIACAT.H） ……（224）

6.2.5 TWindowsObject类 (WINDOBJ.H) …… (226)
6.2.6 TWindow类 (WINDOW.H) …… (229)
6.2.7 TDialog类 (DIALOG.H) …… (232)
6.3 消息及响应 …… (235)
6.3.1 输入机制 …… (235)
6.3.2 标准的消息循环 …… (236)
6.3.3 OWL消息循环函数 …… (237)
6.3.4 OWL消息响应函数 …… (238)
6.3.5 默认消息处理 …… (239)
6.3.6 消息的分类 …… (240)
6.4 在主窗口中对消息的响应 (OWFCS-01.CPP) …… (243)
6.4.1 功能要求 …… (243)
6.4.2 程序设计 …… (243)
6.4.3 程序清单 …… (243)
6.4.4 程序说明 …… (245)
6.5 在主窗口中显示字符信息 (OWFCS-02.CPP) …… (246)
6.5.1 功能要求 …… (246)
6.5.2 程序设计 …… (246)
6.5.3 程序清单 …… (246)
6.5.4 程序说明 …… (248)
6.6 绘制图形的输出 (OWFCS-03.CPP) …… (248)
6.6.1 功能要求 …… (248)
6.6.2 程序设计 …… (249)
6.6.3 程序清单 …… (250)
6.6.4 程序说明 …… (255)
6.7 菜单 (OWFCS-04.CPP) …… (255)
6.7.1 功能要求 …… (256)
6.7.2 菜单资源 …… (256)
6.7.3 程序设计 …… (257)
6.7.4 程序清单 …… (258)
6.7.5 程序说明 …… (262)
6.8 图形文件 (OWFCS-05.CPP) …… (263)
6.8.1 功能要求 …… (263)
6.8.2 程序设计 …… (264)
6.8.3 程序清单 …… (264)
6.8.4 程序说明 …… (272)
6.9 弹出帮助窗口 (OWFCS-06.CPP) …… (273)
6.9.1 功能要求 …… (273)
6.9.2 程序设计 …… (273)
6.9.3 程序清单 …… (281)

第七章 ObjectWindows的应用

7.1 ObjectWindows应用综述 …… (283)
7.1.1 ObjectWindows的继承体系 …… (283)
7.1.2 关于Windows API函数 …… (285)
7.1.3 Windows消息 …… (288)
7.1.4 默认的消息处理 …… (290)
7.1.5 发送消息 …… (290)
7.1.6 消息值域 …… (291)
7.1.7 用户定义的消息 …… (291)
7.2 应用程序 (OWAPP-11.CPP) …… (292)
7.2.1 应用程序流程 …… (292)
7.2.2 初始化应用程序 …… (293)
7.2.3 运行应用程序 …… (295)
7.2.4 关闭应用程序 …… (295)
7.3 界面对象(OWAPP-12.CPP) …… (296)
7.3.1 TWindows Object …… (296)
7.3.2 为何要用界面对象 …… (296)
7.3.3 窗口的父子关系 …… (297)
7.3.4 消息处理 …… (298)
7.4 窗口对象 (OWAPP-13.CPP) …… (302)
7.4.1 TWindow类 …… (302)
7.4.2 初始化和创建窗口对象 …… (302)
7.4.3 窗口类登记 …… (306)
7.4.4 滚翻窗口 (OWAPP-14.CPP) …… (309)
7.4.5 编辑窗口 (OWAPP-15.CPP) …… (314)
7.4.6 文件窗口 …… (318)
7.5 对话框对象 (OWAPP-16.CPP和OWAPP-17.CPP) …… (318)
7.5.1 使用对话框资源 …… (318)
7.5.2 使用子对话框对象 …… (318)
7.5.3 输入对话框 …… (331)
7.5.4 文件对话框 …… (334)
7.6 控制对象 …… (338)
7.6.1 控制对象的使用 …… (338)
7.6.2 控制焦点与键盘 …… (341)
7.6.3 列表框控制 (OWAPP-18.CPP) …… (341)
7.6.4 组合框 (OWAPP-19.CPP) …… (346)
7.6.5 静态控制 (OWAPP-20.CPP) …… (351)
7.6.6 编辑控制 (OWAPP-21.CPP) …… (353)
7.6.7 按钮控制 …… (360)

7.6.8 复选框和单选按钮 …… (360)
7.6.9 成组框 (OWAPP-22.CPP) …… (362)
7.6.10 滚动条 (OWAPP-23.CPP) …… (365)
7.6.11 传送控制数据 (OWAPP-24.CPP及OWAPP-25.CPP) …… (370)
7.7 定制的控制对象 …… (380)
7.7.1 修改预定义控制 (OWAPP-26.CPP) …… (380)
7.7.2 定制控制 …… (386)
7.8 MDI对象 …… (388)
7.8.1 MDI应用程序的组成 …… (388)
7.8.2 构造MDI窗口 …… (389)
7.8.3 MDI应用程序中消息处理 …… (391)
7.8.4 管理MDI子窗口 …… (391)
7.8.5 程序例子 (OWAPP-27.CPP) …… (391)
7.9 流对象 …… (395)
7.9.1 iostream流库 …… (396)
7.9.2 重载操作符≪和≫ …… (396)
7.9.3 流类和TStreamable …… (397)
7.9.4 流管理器 …… (397)
7.9.5 流类的构造函数 …… (399)
7.9.6 流类的名称 …… (400)
7.9.7 使用流管理器 …… (400)
7.9.8 流集合 (Collections on streams) …… (402)

第八章 ObjectWindows应用设计

8.1 标准应用程序设计要求 …… (406)
8.1.1 设计标准 …… (406)
8.1.2 标准的外观和性能 …… (407)
8.1.3 编写安全程序 …… (409)
8.2 编辑类实用程序 (OWEDIT40.CPP) …… (413)
8.2.1 功能要求 …… (413)
8.2.2 程序设计 …… (414)
8.2.3 程序项目文件和程序清单 …… (418)
8.3 工程应用程序界面设计 …… (421)
8.3.1 功能要求 …… (421)
8.3.2 程序设计 …… (422)
8.3.3 程序项目文件和程序清单 …… (431)
8.4 动态数据交换程序(DDE) 程序设计 …… (440)
8.4.1 DDE概述和功能要求 …… (444)
8.4.2 程序设计 …… (448)
8.4.3 程序项目文件和程序清单 …… (459)

8.5 GDI应用程序 (OWGDITO.CPP) …… (473)
8.5.1 GDI概述与功能要求 …… (474)
8.5.2 程序设计 …… (485)
8.5.3 程序项目文件和程序清单 …… (493)
附录A GREP文本搜索应用程序 …… (504)
附录B Borland C++3.1的新特点 …… (509)
附录C Turbo Vision类的继承体系 …… (510)
附录D ObjectWindows继承体系 …… (516)
附录E 应用问答 …… (518)
参考文献 …… (522)

第一章　界面设计导论

用户界面设计已历经了几代的发展。

第一代，在80年代中期，行式界面是界面最贫乏的时代。用户输入数据行和命令行，计算机告诉用户的也是逐行的信息，如EDLIN行编辑，BASIC语言，DEBUG，dBASE II／III。为终端用户开发的软件也是这样的界面，目前仍有在使用的。我们称行式界面为一维界面。

第二代，在80年代末，菜单和窗口兴起，使得用户对软件系统能看到了全貌：了解当前运行到什么地位，如何进退，並且同时能看到不同部分的信息。典型的软件有TURBO C，PCTOOI ^等。这种平面式界面，我们称它为二维界面。

第三代90年代初，Windows的出现，除了其他优越的特点外也使界面出现重大的改革。呆板的“死”的界面“活”了起来，用户可以任意改变各窗口的大小、位置，窗口叠了起来，有了层次，立体化了。我们称为三维界面。典型例子，是Borland C＋＋3.0／3.1，Windows 3.0／3.1及在其上运行的产品。另一个特点是一些命令的图符化，用户可以“感觉”到如何操作，所以也是直觉的三维界面。

第四代，90年代中由于兴起多媒体技术，计算机与人的界面，已不单单是视觉，在视觉上也不单单是静止的（或者是动画式—变动的静止画面）出现了“电视式”的界面，声音的交互，还可能出现触觉的交互。可称为超三维界面。

当前大量的终端用户的应用界户还处于第一代，第二代阶段。软件界在积极开发第三代界面。

但是，第三代界面的设计，由于其复杂性，往往占用了软件人员大量的精力，甚至比处理专业任务还要多。更为悲惨的是，一当用户要求有变化，任务有变化，或者新的任务的提出，程序员不得不从源程序级重新开始。譬如，用C语言设计的界面，无法解决上述的困难。

出路在那里？首先是语言工具转向到面向对象语言—C＋＋。利用该语言继承、封装和多态性等特点可以部分解决上述困难。但是，处理界面的技术很烦杂，用户自己设计一套经得起考验的软件，要费大量精力和时间。更简捷的道路，是利用好的商品软件工具。在Borland C＋＋商品软件中的Turbo Vision（简称TV）及ObjectWindows（简称OW）是属于最理想的工具。但是它们不是工具箱，不像TURBO C的TOOLS，TURBO PROLOG的TOOLBOX。一般工具箱是各种工具的集合，各个工具是相互独立的。使用容易且很快熟悉，并应用它们。而TV和OW是两个复杂、庞大的工具，我们先要花费精力，熟悉它们，但一旦拥有，则其乐无穷。

我们拥有自行车—TURBO C及其TOOLS工具等，我们也拥有汽车—BORLAND C＋＋及其工具TV及OW。学会骑自行车是容易的，但学会开汽车就要经过学习班、实习期等，要花费多得多的精力。但是既然有了汽车，谁不想去开开呢？

1.1 Windows的影响

自从Windows出现后，它的优异特性使程序员和用户为之倾倒。但是Windows与传统软件几乎全异的概念、复杂的软件开发工具、大量的应用程序接口的库函数又使软件人员望而生畏。现在Borland C++中两个软件工具包（ObjectWindows及Resource Workshop）极大地减轻了软件人员的负担。因此可以预见，Windows的应用软件将会加速向市场推出。

而Windows的多窗口，标准的资源模式及消息驱动机制也影响到传统的DOS应用程序。Borland C++中Turbo Vision工具将帮助软件人员开发的DOS程序具有一种“Windows”的味道。在我们用Turbo Vision和ObjectWindows工具开发了应用软件后，再回头看看现在还在使用的传统软件，似乎已有“隔世之感”。

1.1.1 Windows应用特点

（1）DOS与Windows的交互作用

Windows提供了许多操作系统功能，Windows应用程序与Windows系统本身有着非常紧密的相互作用。应用程序的运行，实际上是不断地与DOS操作系统及Windows操作系统的相互作用的过程。见图1.1-1的示意图。

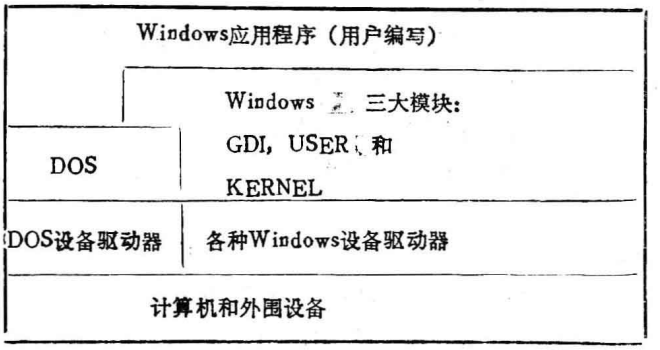


图1.1-1 Windows应用程序与DOS操作系统和Windows操作系统的相互作用

在运行时刻，将调用三个动态连接库（DLL），当系统加载时，将把DLL与应用程序连接在一起，因而尽可能地减少每个程序的代码量，三个主要函数库的内容如下：

函数库	功能
GDI	提供图形设备接口。
Kernel	提供系统服务，如多任务，存储管理和资源管理等。
User	提供窗口管理功能，管理整个Windows环境和应用程序窗口。

（2）Windows应用程序的特点：

① .EXE文件格式与DOS的不同，是专门在Windows下运行的。

②一般在屏幕上的一个矩形窗口内，或缩小为图标中运行。

③应用程序的显示和功能执行遵循“标准的”，格式一致的用户界面原则。

④可以同时与其它的Windows应用程序或非Windows应用程序一起运行。

⑤能够和其他Windows应用程序通信和共享数据。

（3）开发Windows程序的特点：

①设备无关的图形保证图形应用程序可以运行于多种标准显示适配器。

②直接支持各种类型的打印机，显示器和鼠标器等。

③提供更多的内存。

④支持菜单、图标和位图等。

⑤丰富的图形例程库。

（4）用户使用Windows应用程序特点：

①标准的操作，因为所有应用程序的界面是标准化的，一致化的。

②统一的设备驱动程序。

③互相协作与通讯。

④能同时运行多个应用程序。

⑤访问更多内存。

1.1.2 与DOS完全不同的四个概念

（1）消息驱动编程

传统的编程方式是过程式的关联的驱动。程序从开始到结束都被定义好了。例如，图1.1-2中列出一个过程式的程序：进行合同数据的登录。

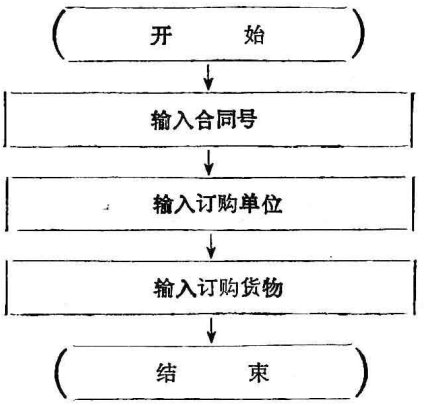


图1.1-2　一个过程式的关联驱动程序

现在考虑一个现实情况：如果正在输入订购单位时，觉得应该查一下该单位的信誉度。如果原来程序没有设计这种意外情况，（即使要设计出来，也有重重困难），那么就要设计一层一层退出菜单，再进入有关信誉度的数据文件，查阅后，重新回到原来的合同登录程序。

如果从另外一个角度——事件驱动方式去处理，上述极困难的问题一下就不成为问题，而是轻而易举的事情。在Windows下所有的事件产生消息。消息是关于用户接口的一些改变信息。例如，单击一次鼠标，或者一个击键。对程序员来说，消息是一个16位的无符号值，并赋予它一个符号常量。那些常量都以WM_（Windows Message）字符开始。后面跟上英文的词汇，有些是全词，有些是缩写，有些是词首。但全部都是大写，我们习惯以后，是容易

“看懂”的。如：

WM_LBUTTONDOWN	鼠标左（L）键按下
WM_KEYUP	击键已释放
WM_NCMOUSEMOVE	非用户区（None Custom_NC）鼠标移动
WM_ACTIVATE	报告该窗口是活动的
WM_CLOSE	要求关闭窗口
WM_GETTEXT	取回正文
WM_HSCROLL	水平（H）滚动条被选中

在编写Windows程序中，要处理的事是确定哪些消息应处理和哪种消息应忽略。一开始进入到这种新机制编程是不习惯的，但是尝到味道后，就感到传统的应用程序太束缚手脚，简直无法使用，而TV作为提供开发DOS应用程序的工具，正是以事件驱动方式为基础的，因此它是很受欢迎的。

（2）多任务与程序调度

Windows是一个多任务系统。用TV开发的DOS应用程序也可以是一个多任务的。

多任务，是指几个程序在“同一时刻”运行。一个CPU下，通过调度，使某一个程序在一小段时间运行。在传统的操作系统里，调度由时钟来完成：每个程序被分配给一个“时间片段”。通过中断，切换到另一个程序。这是一种由操作系统中断的优先调度。

而以消息驱动的程序中，不用优先调度，程序不能被操作系统中断，而是每个程序通过消息处理自动中断操作，以使其它程序运行，调度系统是建立在消息传送机制之上，当一个程序完成处理一条消息时，又申请另一条消息。在这里，用户的操作或其他程序发出的消息是决定调度的因素。

消息提供了程序的输入。而这类软件系统的输出又有自己的特点。在TV下是特殊的文本输出。在Windows下是图形输出。

（3）输出

TV创建的应用程序是在文本方式下的输出，但是这种文本的输出有其自己的特殊性，往往经过类的成员函数来处理，而不是调用C／C++的标准输出库函数。在TV下可以切换到DOS的图形状态，但是由于要返回到原应用程序的事件管理，因此程序也要经过专门的安排。详见第二章到第四章。

而Windows程序所有的输出都是图形，因此有几个新的特色：

①文本将作为图形处理，因此使文本输出更显困难。

②因为文本也是图形，所以正文和图形混合编排的应用是很方便的事。

③Windows有一个图形驱动接口GDI，它是独立于具体设备的，因此有很强的通用性。Windows程序能经过同样的调用，输出到不同的设备中去：如屏幕、打印机和绘图仪等。GDI可以识别四种不同类型的设备：

①屏幕；

②硬拷贝设备（如打印机、绘图仪）；

③位图；

④元文件。

其中③和④是伪设备，即一种在RAM里或在磁盘里存储图像的方法，包括在几个应用程序之间图像共享的标准规约。