

软件高级编程实例精解丛书

高级编程实例精解

Visual C++

谢歆 赵国华 编著 雨人 策划



国防工业出版社

Visual C++ 高级编程实例精解

谢歆 赵国华 编著

雨人 策划

国防工业出版社

• 北京 •

图书在版编目 (CIP) 数据

Visual C++高级编程实例精解/谢歆等编著. —北京:
国防工业出版社, 2001.7

(软件高级编程实例精解丛书)

ISBN 7-118-02460-0

I. V... II. 谢... III. C 语言-程序设计
IV. TP312

中国版本图书馆 CIP 数据核字 (2000) 第 87285 号

国防工业出版社出版发行

(北京市海淀区紫竹院南路 23 号)

(邮政编码 100044)

北京奥隆印刷厂印刷

新华书店经售

*

开本 787×1092 1/16 印张 34 $\frac{1}{2}$ 800 千字

2001 年 7 月第 1 版 2001 年 7 月北京第 1 次印刷

印数: 1—3000 册 定价: 49.00 元

(本书如有印装错误, 我社负责调换)

前 言

Microsoft 公司的 Visual C++ 是世界上最优秀的面向对象编程环境之一。其每一个新版本的推出都受到人们的关注和广大程序员的欢迎。它的以 MFC 应用框架为基础的编程方法将编程环境提供的代码和资源编辑器、编译器、连接、调试器、AppWizard、ClassWizard、Browser 等不同编程阶段使用的工具天衣无缝地结合在一起，大大提高了代码编制的自动化程度，使得编程工作简洁而高效。

融入精彩纷呈的实例

在编程语言学习中，优秀的实例程序的作用是极其重要的。单纯的讲解或者单纯的演示性的实例往往不够深刻，面临实际开发时还是会遇到这样那样的问题难以解决。基于此点，本书将通过一系列精彩纷呈、令人印象深刻的典型实例，讲述使用 Visual C++ 开发 Windows 应用程序的各项高级技术。具体来说，本书每个专题都将实现一个较大规模并具有一定实际意义的示范程序，当然，书中不乏一些短小精悍的实例，来说明主要实例无法说明的关键性问题。这些精彩纷呈的实例程序，将使你能够透彻地掌握 Visual C++ 编程技术及技巧，快速成为高级开发者。

体验高级开发的乐趣

Visual C++ 以其最能体现编程者的风格、编程效率高特性，为广大程序员所钟爱。除了秉承以前版本的各项优点以外，还大大扩充了 MFC 的内容，进一步扩展网络编程功能，有效改善各工具的集成性能。

本书通过一系列出色的实例程序深入浅出地讲解 Visual C++ 编程的方方面面。书中所有主要实例均注意范例程序的实用性、典型性与趣味性，并且具有很强的应用背景。本书的各个章节将详细分析这些实例程序的开发，并围绕实例讲述关键知识点和编程技术。通过精心设计的精彩范例，你将轻松地学习并精通 Visual C++ 编程，体验到高级开发的乐趣。

记住，您是未来的高级开发者

本书主要是为那些对 Visual C++ 编程有所了解并想学习高级技术和技巧的读者所编写，同时也是很有价值的软件开发人员的参考资料。本书不是 Visual C++ 入门参考书，如果你是 Visual C++ 的初学者，本书可能并不适合你。

最好地发挥该书的作用

本书共有 11 章，深入地介绍了 Visual C++ 在各个领域的编程技术。

第1章从编程思想与理念方面对 Visual C++ 进行系统的分析, 介绍了 Microsoft 在 Visual Studio 7.0 中新推出的开发语言 C# 及其与 C/C++ 的异同, 以及为推动新一代因特网的发展而提出的 .NET 战略, 同时深入讨论 Visual C++ 的调试技术。

第2章讲述了图形设备接口与图形数据结构。通过一个图形系统的框架搭建, 将图形设备接口与图形数据结构的有关技术充分地展现出来。

第3章实现了图形系统程序的完善, 并通过整个实例的实现, 具体讲述消息映射与处理的过程, 以及控制条使用的实际操作。同时, 作为图形系统的深入部分, 对图形的放大与移动进行了分析与讲述。

第4章介绍了对话框技术, 分析了对话框的编程理论, 并对控制的使用作了详细的讲解。我们并没有将精力放在简单的模式对话框上, 而是通过将非模式对话框与卡片式对话框两种对话框技术融入图形系统, 力图使读者对对话框技术有深入的认识与掌握。

第5章是多任务处理, 包括多进程和多线程两个方面。多进程部分介绍了进程的启动、终止、优先级和进程间通信的实现; 多线程部分则包括线程的创建和终止、线程优先级、线程同步和线程间通信等内容。

第6章讲述了 Visual C++ 多媒体开发技术, 并就其使用最广泛的媒体控制接口作了深入的讨论。我们将通过对一个精彩范例——波形音频播放器的定制与分析, 讲述 Visual C++ 的多媒体编程技术, 而音频编程作为重点介绍得非常深入透彻。

第7章是 Visual C++ 数据库开发部分。首先就数据库的基础知识与 SQL 语言作了简要但是必要的概述, 而后详细讨论了 MFC ODBC 编程技术, 包括如何应用 MFC ODBC 类库为数据库编程服务以及 RFX 机制与 BULK RFX 机制, 并示范了如何创建数据库与管理数据源, 最后通过一个实用的公司雇员信息管理系统的范例程序, 展示了数据库编程技术的实际应用与编程技巧。

第8章详细阐述了 Windows Sockets 的相关概念、技术规范、编程原理以及广播技术。其中着重介绍了 MFC 对 Windows Sockets 的支持, 最后以一个多线程的服务器程序为例来阐述如何应用 Windows Sockets 技术进行网络应用程序的编写。

第9章着重介绍 ActiveX 控件的相关知识、创建及使用等内容。具体包括 ActiveX 技术的发展概述, ActiveX 控件的特点及设计开发过程, ActiveX 脚本技术简介, ActiveX 控件的压缩和签名, 以一个时钟控件来说明 ActiveX 控件的编制过程及 ActiveX 控件在应用程序和 Web 网页中的使用。

第10章是对 ActiveX 文档的介绍, 介绍了 ActiveX 文档的概念, 使用 ActiveX 文档的优点和方法, 以及 ActiveX 文档的特性、结构等内容, 并讲述了如何实现一个 ActiveX 文档服务器和文档容器程序。

第11章介绍 ISAPI 的原理及与传统 CGI 程序的区别, MFC 中的 ISAPI 类及 PARSEMAP 宏的使用, ISAPI 扩展和 ISAPI 过滤器的概念、区别及其编程过程, 本章最后给出了一个在 Web 网页上进行会员注册的 ISAPI 扩展的应用实例。

附带光盘价值多多

本书所附带的光盘(不随书发售)包括了所有实例的源程序代码, 并且在代码中给出了详尽的注释。认真地阅读分析这些源代码, 你将体会到各种功能的实现思路和方法, 相信

一定会有较大的收获。需要光盘的读者可从<http://www.ndip.com.cn/computer> 下载或发 email 至 zfxin@ndip.com.cn 索取。

使用本书的几个约定



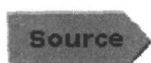
标示提醒读者应该注意的文字。



标示对读者具有一定提示和启发性的文字。



标示对书中某些内容作补充说明的文字。



标示书中出现的源代码。

目 录

第 1 章 MFC、C#与 Microsoft.NET	1
1.1 Windows 程序原理	3
1.1.1 Windows 程序工作原理	3
1.1.2 消息循环与输入	4
1.1.3 面向对象技术与 Windows 编程	5
1.2 一种新型开发语言——C#	6
1.2.1 为什么要使用 C#	7
1.2.2 C#和 C/C++有哪些主要的不同	10
1.3 NWGS 运行时基础	21
1.3.1 几个基本概念	22
1.3.2 虚拟对象系统	23
1.4 Microsoft .NET 战略	25
1.4.1 Microsoft .NET 战略的核心内容	25
1.4.2 Microsoft .NET 的特色	26
1.4.3 Microsoft .NET 的组成	26
1.4.4 Microsoft .NET 的意义	29
1.5 Microsoft .NET 开发	29
1.5.1 网络服务一览	29
1.5.2 Microsoft .NET 框架	30
1.6 MFC 程序结构剖析	37
1.6.1 MFC 类库	37
1.6.2 程序结构框架	38
1.7 Visual C++程序调试	40
1.7.1 常规调试技术	40
1.7.2 高级调试技术	46
1.7.3 异常处理	49
1.7.4 内存漏洞的检查	52
第 2 章 图形设备接口与图形数据结构	55
2.1 设备上下文	57
2.1.1 基本数据类型	57

2.1.2 设备环境类	59
2.1.3 设备环境类的内容	61
2.2 绘图对象	61
2.2.1 GDI 绘图对象的分类	61
2.2.2 绘图对象的使用	62
2.2.3 画笔 CPen	64
2.2.4 画刷 CBrush	65
2.2.5 字体 CFont	67
2.2.6 其它图形对象	69
2.3 坐标系与映射模式	71
2.4 绘图状态管理与绘图操作	73
2.4.1 绘图状态管理	73
2.4.2 绘图操作	75
2.5 图形数据的存储状态	80
2.6 实例图形系统的数据结构与存储组织	81
2.6.1 直线的存储组织	82
2.6.2 连续直线和多边形区域的存储组织	86
2.6.3 圆和圆形区域的存储组织	91
2.6.4 文本的存储组织	94
2.7 图形系统的建立	96
2.7.1 坐标系的建立	96
2.7.2 各种图形元素的绘制	98
2.7.3 实现图形的绘制	104
第 3 章 消息处理与控制条	107
3.1 消息映射与消息处理	109
3.1.1 Windows 消息	109
3.1.2 消息的发送、接收与传递	111
3.1.3 消息映射	112
3.2 利用鼠标绘制图形	114
3.3 控制条的使用	116
3.4 实现完整的图形绘制系统	121
3.4.1 全局变量问题	121
3.4.2 实现鼠标作图	122
3.4.3 图形的放大与移动	136
3.4.4 工具条与状态条的使用	142
第 4 章 对话框技术	145

4.1 对话框概述	147
4.1.1 对话框的组成与分类	147
4.1.2 常用的对话框类	148
4.2 控制的使用	154
4.2.1 Windows 控件概述	154
4.2.2 各种控制的使用方法	156
4.3 模式对话框	172
4.4 实例图形系统功能扩展 —— 使用非模式对话框	173
4.4.1 非模式对话框的特点	174
4.4.2 窗口对象的自动清除	175
4.4.3 实现非模式对话框	176
4.5 实例图形系统功能扩展 —— 使用卡片式对话框	193
4.5.1 卡片式对话框概述	193
4.5.2 卡片式对话框的创建	193
4.5.3 卡片式对话框的运行机制	194
4.5.4 实现卡片式对话框	195
第 5 章 多任务处理	215
5.1 进程的创建	217
5.1.1 进程的启动和终止	217
5.1.2 进程的优先级	220
5.1.3 多进程实例	221
5.2 进程间通信	223
5.2.1 使用内存映射文件	223
5.2.2 利用共享内存动态链接库	225
5.2.3 用于传输只读数据的 WM_COPYDATA	225
5.2.4 调用 ReadProcessMemory() 和 WriteProcessMemory() 函数	226
5.2.5 进程间消息的发送与接收	227
5.3 线程的创建	228
5.3.1 基于 Win32 API 的多线程	228
5.3.2 基于 MFC 的多线程	230
5.4 辅助线程和用户界面线程	231
5.4.1 创建辅助线程	232
5.4.2 创建用户界面线程	233
5.4.3 终止线程	235
5.4.4 线程的优先级	236
5.5 线程的同步	237
5.5.1 线程同步的必要性	237
5.5.2 等待函数	239

5.5.3 线程同步对象	240
5.6 线程间通信	243
5.6.1 使用全局变量通信	243
5.6.2 使用自定义消息通信	244
5.6.3 使用事件对象通信	246
5.7 多线程应用实例	248
5.7.1 创建对话框界面	249
5.7.2 生成线程类	249
5.7.3 在对话框类中添加代码	255
第 6 章 Visual C++多媒体开发	265
6.1 Visual C++与多媒体开发	267
6.1.1 多媒体概述	267
6.1.2 多媒体程序设计	269
6.1.3 多媒体文件	275
6.2 多媒体的组成	276
6.2.1 静态图形	276
6.2.2 动画技术	280
6.2.3 音频技术	283
6.3 媒体控制接口	284
6.3.1 命令字符串方式	285
6.3.2 命令消息方式	288
6.4 波形音频技术	291
6.4.1 波形音频结构	291
6.4.2 波形音频的信息获取	292
6.5 波形音频的处理	295
6.5.1 高级音频函数	295
6.5.2 利用 MCI 处理波形音频	297
6.5.3 低级设备函数	299
6.6 定制波形音频播放器	300
6.6.1 建立波形音频类	300
6.6.2 创建应用程序框架	306
6.6.3 程序功能实现	309
第 7 章 Visual C++数据库开发	323
7.1 Visual C++与数据库开发	325
7.1.1 数据库知识简介	325
7.1.2 Visual C++的数据库开发工具	327

7.1.3 Visual C++数据库开发的特点	328
7.2 SQL 语言	329
7.2.1 SQL 概述	329
7.2.2 SQL 处理技术	330
7.2.3 SQL 语言	333
7.2.4 SQL 查询	335
7.3 MFC ODBC 概述	338
7.4 MFC ODBC 数据库类	340
7.4.1 CDatabase 类	341
7.4.2 CRecordset 类	346
7.4.3 CRecordView 类	351
7.4.4 CFieldExchange 类	354
7.5 RFX 机制与 Bulk RFX 机制	355
7.5.1 RFX 机制	355
7.5.2 深入记录字段交换过程	356
7.5.3 Bulk RFX 机制	358
7.6 创建数据库	359
7.6.1 创建数据库的一般步骤	359
7.6.2 添加表项	361
7.7 数据源管理	363
7.8 记录集管理	366
7.8.1 记录集的组员	366
7.8.2 过滤记录集	367
7.8.3 排序记录集	368
7.8.4 参数化记录集	368
7.8.5 多表联合查询	370
7.9 公司雇员信息管理系统实例剖析	371
7.9.1 建立数据库应用程序框架	372
7.9.2 应用程序界面设计	382
7.9.3 显示记录信息	383
7.9.4 多表参数化查询	385
7.9.5 操作数据库记录	389
第 8 章 应用 Windows Sockets 编程	393
8.1 Windows Sockets 概述	395
8.1.1 Sockets 基本概念	395
8.1.2 MFC 中的 Sockets	398
8.2 Windows Sockets 编程原理	400
8.2.1 基于 API 的编程模式	401

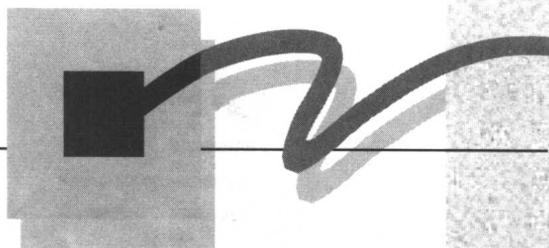
8.2.2 基于 MFC 的编程模式	407
8.3 Windows Sockets 广播技术	410
8.3.1 建立一个可以广播的套接字	410
8.3.2 发送广播消息	411
8.3.3 接收广播消息	412
8.4 多线程 Windows Sockets 编程实例	414
8.4.1 加入 Windows Sockets 支持	415
8.4.2 添加新类	415
8.4.3 改写相关文件	415
8.4.4 编写客户端程序	429
第 9 章 创建和使用 ActiveX 控件	433
9.1 ActiveX 技术概述	435
9.1.1 从 OLE 到 ActiveX	435
9.1.2 ActiveX 技术的内容	436
9.2 ActiveX 控件	438
9.2.1 从 COM 角度理解 ActiveX 控件	438
9.2.2 ActiveX 控件的设计开发过程	439
9.2.3 几种典型控件的实现	443
9.2.4 几种特殊控件属性的实现	445
9.3 ActiveX 脚本技术	447
9.3.1 ActiveX 脚本的背景	448
9.3.2 ActiveX 脚本结构	449
9.4 ActiveX 控件的压缩和签名	453
9.4.1 ActiveX 控件的压缩	454
9.4.2 ActiveX 控件的签名	456
9.5 创建 ActiveX 控件	457
9.5.1 显示 ActiveX 控件界面	457
9.5.2 增加属性、方法和事件	465
9.6 ActiveX 控件的使用	472
9.6.1 在 Web 页面上使用 ActiveX 控件	472
9.6.2 ActiveX 控件在应用程序中的使用	475
第 10 章 创建 ActiveX 文档应用程序	479
10.1 ActiveX 文档概述	481
10.1.1 什么是 ActiveX 文档	481
10.1.2 创建 ActiveX 文档链接	484
10.1.3 ActiveX 文档特性	485

10.2 ActiveX 文档结构	486
10.2.1 ActiveX 文档实现的接口简介	486
10.2.2 ActiveX 文档实现的特殊接口	487
10.3 实现 ActiveX 文档应用程序	491
10.3.1 MFC 库中实现 ActiveX 文档的机制	491
10.3.2 实现 ActiveX 文档服务器程序	493
10.3.3 实现 ActiveX 文档包容器程序	502
第 11 章 ISAPI 编程技术	505
11.1 CGI 与 ISAPI	507
11.1.1 传统 CGI 原理与性能	508
11.1.2 ISAPI 原理与性能	508
11.2 MFC 中的 ISAPI 支持	510
11.2.1 MFC 中的 ISAPI 类	510
11.2.2 MFC 中的 PARSEMAP 宏	512
11.3 ISAPI 扩展	514
11.3.1 使用 ISAPI 扩展向导	515
11.3.2 客户端输入	518
11.4 ISAPI 过滤器	519
11.4.1 ISAPI 过滤器介绍	519
11.4.2 使用 ISAPI 过滤器向导	521
11.4.3 创建 ISAPI 过滤器程序	522
11.5 用 ISAPI 扩展编写用户注册程序	524
11.5.1 增加命令处理函数	524
11.5.2 增加数据库支持	532
11.5.3 增加线程安全控制	538

第一章

MFC、C#与 Microsoft .NET

Windows 程序原理
C#开发语言简介
Microsoft .NET 战略
MFC 程序结构剖析
Visual C++ 程序调试



本

章

导

读

同其它编程工具相比, Visual C++在提供可视化编程方法的同时, 也适于编写直接对系统进行底层操作的程序。Visual C++所提供的 Microsoft 基础类库 MFC 对 Windows/NT 所用的 Win32 应用程序接口进行了十分彻底的封装, 大大节省了应用程序的开发周期, 降低了开发成本, 也使得 Windows 程序员从大量的复杂劳动中解脱出来。

本章从 Windows 的程序原理入手, 在对之深入了解的基础上, 进一步剖析 MFC 程序的结构, 使得我们可以从一个相对较高的层次分析 Visual C++的理论问题。同时, 考虑到如果要对 Visual C++编程达到精通的程度, 熟练而且全面的调试技术是必不可少的, 因此在本章最后详细讨论了 Visual C++的调试技术。

在即将推出的 Visual Studio 7.0 中, Microsoft 根据其 .NET 战略提出了一种新的开发语言 C#, 通过 C#开发人员可以快速地建立大范围的基于 Windows 网络平台的应用程序。C#提供了大量的开发工具和服务以帮助开发人员开发基于计算和通信的各种应用程序。

本章内容包括:

- Windows 程序的工作原理以及它与面向对象技术的结合。
- C#编程语言简介, 着重介绍其特点及与 C++的不同。
- Microsoft .NET 战略的内容、组成、意义及其开发过程。
- 对 MFC 类库的概述, 以及对 MFC 程序结构的剖析。
- 对 Visual C++进行调试的常规技术及高级技术, 包括异常处理与内存泄漏的处理。

1.1 Windows 程序原理

在过去,还没有现在的这些设计精美的应用程序开发工具的时候,进行 Windows 程序设计是一件痛苦异常的事情。而在今天,一个对 Windows 程序运行的内部机制几乎一无所知的初入门者,只需要通过不到一天的学习,也可以使用如 Visual Basic 之类的程序开发工具创建出功能完整的 Windows 应用程序,这在几年前还是一件不可思议的事。

在 Microsoft 的另一种易学易用的编程工具 Visual Basic 中,从某种角度说,Windows 程序不是编出来的,而是由程序员画出来的。但是我们必须清楚认识到,一个出色的 Windows 应用程序并不仅在于在屏幕上绘出程序的各个窗口和在窗口中恰当地安排每一个控件。对于具有一定基础的程序员而言,更重要的内容在于知道 Windows 和 Windows 应用程序的运行机制,以及它们之间以何种方式来进行通信,然后,明确自己在编写 Windows 程序时所需做的工作。换句话说,我们需要透过 Windows 漂亮的图形用户界面,认清在底层所发生的每一件事情。然而,真正做到这一点并非是一件容易的事。

虽然通过使用 MFC 和 AppWizard,我们仍可能只需要回答几个简单的问题和添加少数的几条代码就能够生成功能完整的 Windows 应用程序。但是,一个普遍的事实是,没有一个成功的商业软件是使用这样的方式生成的。同时,也只有深入地理解了 MFC 应用程序框架的运行机制,才可能用好和用活这一工具,才能达到熟悉掌握 Visual C++ 的境界。

1.1.1 Windows 程序工作原理

Windows 程序设计是一种完全不同于传统的 DOS 方式的程序设计方法,它是一种事件驱动方式的程序设计模式。传统的 MS-DOS 程序主要采用顺序的、关联的、过程驱动的程序设计方法。一个程序是一系列预先定义好的操作序列的组合,它具有一定的开头、中间过程和结束,程序直接控制程序事件和过程的顺序。这样的程序设计方法不是面向用户的,交互性差,用户界面不够友好,因为它强迫用户按照某种不可更改的模式进行工作,它的基本模型如图 1.1.1 所示。

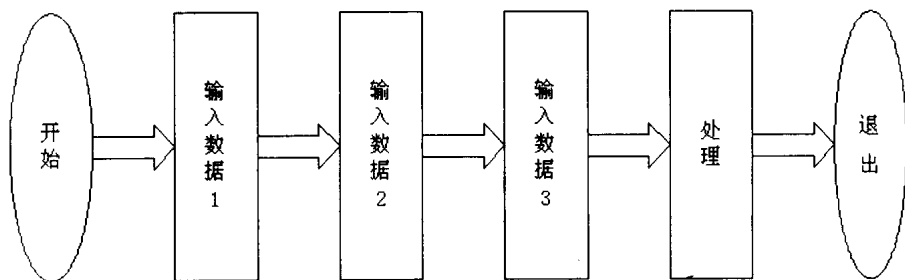


图 1.1.1 过程驱动程序模型

事件驱动程序设计是一种全新的程序设计方法,它不是由事件的顺序来控制,而是由事件的发生来控制,而这种事件的发生是随机的、不确定的,并没有预定的顺序,这样就允

许程序的用户用各种合理的顺序来安排程序的流程。对于需要用户交互的应用程序来说，事件驱动的程序设计有着过程驱动方法无法替代的优点。它是一种面向用户的程序设计方法，它在程序设计过程中除了完成所需功能之外，更多地考虑了用户可能的各种输入，并针对性地设计相应的处理程序。它是一种“被动”式程序设计方法，程序开始运行时，处于等待用户输入事件状态，取得事件后才作出相应反应，处理完毕又返回并处于等待事件状态。它的模型如图 1.1.2 所示，在图中，输入界面 1~3 并没有固定的顺序，用户可以随机选取，以任何合理的顺序来输入数据。

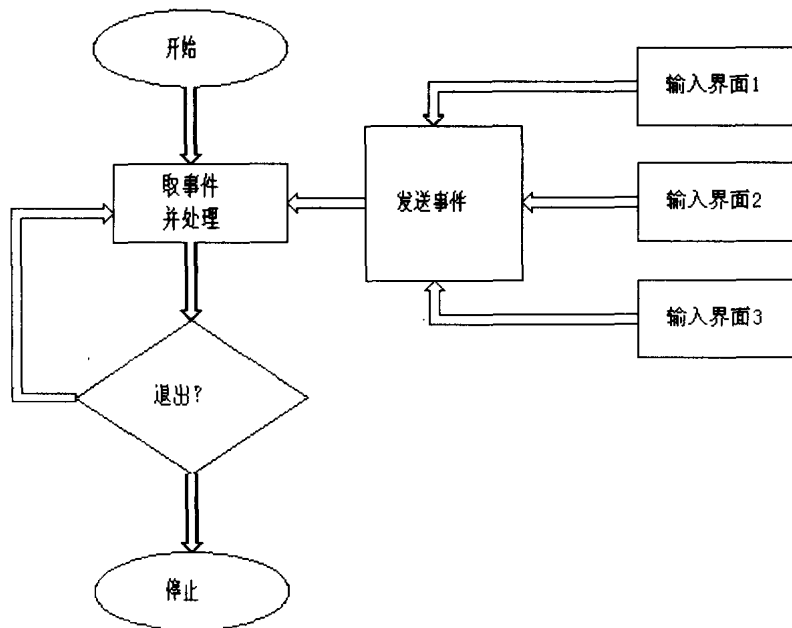


图 1.1.2 事件驱动程序模型

1.1.2 消息循环与输入

消息是一种报告有关事件发生的通知。事件驱动围绕着消息的产生与处理展开，靠消息循环机制来实现。消息类似于 DOS 下的用户输入，但比 DOS 的输入来源要广，Windows 应用程序的消息来源有以下四种：

- 输入消息：包括键盘和鼠标的输入。这一类消息首先放在系统消息队列中，然后由 Windows 将它们送入应用程序消息队列中，由应用程序来处理消息。
- 控制消息：用来与 Windows 的控制对象，如列表框、按钮、检查框等进行双向通信。当用户在列表框中改动当前选择或改变了检查框的状态时发出此类消息。这类消息一般不经过应用程序消息队列，而是直接发送到控制对象上去。
- 系统消息：对程序化的事件或系统时钟中断作出反应。一些系统消息，像 DDE 消息（动态数据交换消息）要通过 Windows 的系统消息队列，而有的则不通过系统消息队列而直接送入应用程序的消息队列，如创建窗口消息。
- 用户消息：这是程序员自己定义并在应用程序中主动发出的，一般由应用程序的