

计算机编程技术丛书

精通

Visual C++ 6.0

刘小石 郑淮 马林伟 杨帆 编著
史惠康 石利文 审校

清华大学出版社
<http://www.tup.tsinghua.edu.cn>

计算机编程技术丛书

精通 Visual C++ 6.0

刘小石 郑 淮 马林伟 杨 帆 编著

史惠康 石利文 审校

清华大学出版社

(京)新登字 158 号

内 容 简 介

Visual C++是 Microsoft 公司推出的可视化开发工具,其最高版本是 6.0,功能十分强大。本书通过大量的实例,深入浅出的介绍了 Visual C++ 6.0 图形界面设计、文件的处理、绘图、数据库、多线程以及网络等方面的程序编制。并且在讲述 VC 的基础上,穿插了大量的编程经验和使用技巧。

全书内容充实,讲解详细,图文并茂,具有一定的可读性。

本书比较适合于使用 VC 编程的初级用户,对专业的程序编制人员也有一定的参考作用。

版权所有,翻印必究。

书 名:精通 Visual C++ 6.0

作 者:刘小石 郑 淮 马林伟 杨 帆

出版者:清华大学出版社(北京清华大学校内,邮编:100084)

<http://www.tup.tsinghua.edu.cn>

印刷者:北京市清华园胶印厂

发行者:新华书店总店北京科技发行所

开 本:787×1092 1/16 印张:30.75 字数:742 千字

版 次:2000 年 2 月第 1 版 2000 年 6 月第 2 次印刷

书 号:ISBN 7-302-01183-4/TP·437

印 数:5001~10000

定 价:46.00 元

《计算机编程技术丛书》出版说明

《计算机编程技术丛书》，包含了现今流行的开发工具系列软件，如 Authorware 5.0、AutoCAD 2000、Visual C++ 6.0、Visual Basic 6.0、Visual FoxPro 6.0 等，是一套易学易用的指导书。既可循环学习，亦可随查随用，使您学有所依，用有所循，快速便捷地掌握软件的操作方法和编程技术，得心应手地解决实际问题。

为了使本系列书有的放矢，力求使其具有如下特点：

知识速查 我们对各编程软件的基本知识做了简要说明，以备速查。使只要有一点软件开发经验的用户就可以在很短的时间内掌握如何编写基本的应用程序。

循序渐进 在编写本书过程中，尽量按照“由浅入深”的顺序来编写，以便于读者的学习。

深入浅出 作者站的角度较高，能深入探讨编程原理及技巧，因为是国内作者所著，能以流利、规范、通俗易懂的语言来说明。

示例详尽 书中的难点、疑点尽量利用实例来进行说明，以便于理解掌握。同时在每个细节中都尽可能地给出详尽的示例，以避免由于一些细微的错误而导致不能掌握某个知识点。

创新思维 以生动的、具有启发性的示例启发读者，且配有具体的步骤和丰富的画面，以及有关的经验提示，让读者深刻领悟该软件的精华。

内容全面 本书所包含的内容丰富而全面（对中级用户而言），以适应读者的需要。

本系列书面向工作在各行各业的计算机编程人员，为他们提供最新的编程技术与编程经验，使他们能尽快跟上技术发展，借鉴他人的经验并结合自身工作，编制出更先进、实用的应用程序。

在本系列书的编著过程中，尽量注意减少冗长无味的说明，代之以具体实用的例子演示。通过例子引导读者把握精髓所在。本书注重开发实例、开发经验、开发技巧和高级特性开发，适应用户的急需。

本系列书是集体智慧的结晶，而且每本书都是由最擅长该开发工具的开发人员执笔。

序

近年来，Windows 操作系统的广泛普及，计算机可视化技术和网络技术的飞速发展，使编程人员产生了对基于 Windows 操作系统的强大开发平台日益强烈的需求。Visual C++ 正是适应这一时代要求的产物。从早期的 Visual C++ 1.5 到现在的 Visual C++ 6.0，每一个版本都是在前一版本的基础上经过不断的完善形成的。

Microsoft Visual C++ 6.0 是 Microsoft Visual Studio 6.0 家族成员之一，是 Visual C++ 的最新版本。它不仅适用于 Windows 编程，还为网络编程、数据库编程提供了强大的支持。与 Visual C++ 5.0 相比，Visual C++ 6.0 具有更快的编译速度、更加友好的编辑界面和更加强大的数据库支持功能。本文将 Visual C++ 6.0 为例，通过大量的实例全面系统地介绍 Visual C++ 6.0 开发环境和编程技术。

本书包括基础篇、提高篇和附录。基础篇向读者介绍了 Visual C++ 编写 Windows 应用程序的基本知识，这些知识是编写 Windows 应用程序所不可缺少的。在具备了基础篇的知识后，提高篇让读者掌握如何使用 Visual C++ 6.0 编写数据库应用程序、多媒体应用程序和网络应用程序等。考虑到初学者的需要，在附录中着重介绍了 C++ 语言中类的概念。同时，附录还给出了 MFC 中常用的类。

本书由中科辅龙计算机技术有限公司抖斗书屋策划，第 1、3、4、8、11、14 章和第 15 章由刘小石编写；第 2、5、10 章和 13 章由郑准编写。第 6、7、9 章和第 12 章由马林伟编写。第 16 章和附录由杨帆编写。全书由石利文和史惠康统稿，徐平负责本书的校排工作。在本书编写过程中，熊长安、乔岷、王毅、刘波、王峰、钟洪亮、王天骄、徐明、黄湘江等提供了部分资料和图片，杨庆涛、赵光明、刘从亮、曹作旺、唐守伟、严娟、唐茹冰、李冰航、蒋方帅和王瑞波参与了部分程序的编制和调试，孙红、杨葆勇、杨建荣等完成了部分录入工作，高华欣、蔡达文、王珂、梁晓林等做了部分审校工作。

在本书编写过程中，李浩群、卢大勇等提供了许多宝贵建议。

抖斗书屋坐落于中科院计算所院内，由中科辅龙计算机技术有限公司领导，是一家拥有雄厚实力的计算机图书创作单位。在本书的编写过程中，书屋的全体员工都付出了大量劳动，借此机会对书屋全体人员的精诚团结表示由衷的感谢！

由于时间仓促、作者水平有限，书中错漏之处在所难免，欢迎读者批评指正。

史惠康

1999 年春于中科院计算所

目 录

第 1 章 Windows 的编程特点	1
1.1 Windows 程序的特点	1
1.1.1 用户图形界面	1
1.1.2 多任务	1
1.1.3 Windows 编程的特点	2
1.1.4 Windows 编程的内部分析	5
1.2 MFC 概述	13
1.2.1 为什么用 MFC 编程	13
1.2.2 MFC 基础类库的优点	14
1.3 小结	15
第 2 章 熟悉 VC++ 6.0 的编程环境	16
2.1 Microsoft Visual Studio 6.0 开发平台	16
2.1.1 标题栏	17
2.1.2 菜单栏	17
2.1.3 工具栏	31
2.1.4 项目、项目工作区和项目工作区窗口	32
2.1.5 源代码编辑窗口	35
2.1.6 输出窗口	35
2.1.7 状态栏	35
2.2 创建一个简单的应用程序	36
2.2.1 MFC AppWizard	36
2.2.2 ClassWizard	36
2.2.3 一个简单的程序	36
2.2.4 源程序清单	45
2.3 小结	56
第 3 章 菜单、工具栏和状态栏	57
3.1 菜单设计	57
3.1.1 创建应用程序的框架	57
3.1.2 打开菜单编辑器	57
3.1.3 修改菜单	58
3.1.4 给菜单联接上快捷键	60
3.1.5 给菜单联接上加速键	61
3.1.6 设置菜单项属性	62
3.1.7 给菜单项联接上代码	64

3.1.8	在程序中改变菜单的属性	66
3.1.9	用灰色显示菜单项	67
3.1.10	用户界面更新消息的处理	68
3.1.11	动态增加菜单	70
3.1.12	删除菜单	73
3.2	用工具栏和状态栏改进程序	74
3.2.1	在新建程序时增加工具栏	75
3.2.2	修改工具栏	76
3.2.3	加入新的工具栏	77
3.2.4	在程序中加入状态栏	79
3.3	小结	81
第 4 章	对话框和控制	82
4.1	认识对话框	82
4.1.1	对话框的组成	82
4.1.2	对话框控件	82
4.1.3	模态对话框和非模态对话框	83
4.1.4	公共对话框	84
4.2	创建对话框	85
4.2.1	创建应用程序的框架	85
4.2.2	创建一个对话框类	90
4.2.3	创建对话框对象, 调用并显示对话框	92
4.2.4	向对话框控件联接数据成员	93
4.2.5	初始化对话框	95
4.2.6	将控件同对话类中相应的消息处理函数联系起来	96
4.3	使用对话框控件	99
4.3.1	对话框控件	99
4.3.2	控件的使用	102
4.4	小结	103
第 5 章	单文档和多文档	104
5.1	文档/视图体系	104
5.1.1	文档和视图	104
5.1.2	文档基类 CDocument	104
5.1.3	视图基类 CView	108
5.2	从单文档界面 (SDI) 到多文档界面 (MDI)	111
5.2.1	单文档界面 (SDI)	111
5.2.2	多文档界面 (MDI)	112
5.2.3	创建多文档应用程序	113
5.3	MFC 提供的视图派生类	120

5.3.1	CScrollView 类	121
5.3.2	源文件	126
5.3.3	CEditView 类	134
5.4	文档模板	136
5.4.1	文档模板的基类 CDocTemplate	136
5.4.2	CSingleDocTemplate 和 CMultiDocTemplate	138
5.4.3	如何创建文档模板	139
5.5	小结	139
第 6 章	文件的读和写	140
6.1	使用 CFile 类向文件写数据和从文件读数据	140
6.1.1	CFile 类	140
6.1.2	CFile 的派生类	146
6.2	用串行化向文件写数据和从文件读数据	147
6.2.1	串行化 (Serialization)	147
6.2.2	创建一个有串行化功能的类	147
6.2.3	CArchive 类	150
6.2.4	对象的串行化操作	153
6.2.5	串行化应用实例	156
6.3	小结	165
第 7 章	鼠标与键盘	166
7.1	使用鼠标	166
7.1.1	鼠标事件和鼠标消息	166
7.1.2	处理用户区鼠标事件	167
7.1.3	处理非用户区鼠标事件	169
7.1.4	鼠标捕捉	171
7.1.5	鼠标画线程序 MyLine	172
7.2	响应键盘	178
7.2.1	键盘事件	178
7.2.2	谁来处理键盘事件	178
7.2.3	处理键盘事件	179
7.2.4	MyKey 测试程序	181
7.3	小结	188
第 8 章	在屏幕上绘图	189
8.1	MFC 绘图函数	189
8.1.1	生成设备情景对象	189
8.1.2	选择绘图工具	190
8.1.3	设置绘图属性	193
8.1.4	绘图	196

8.2	创建 MyDraw 应用程序	201
8.2.1	用 AppWizard 创建 MyDraw 的项目文件	202
8.2.2	编辑应用程序菜单	202
8.2.3	给菜单连接代码	203
8.3	文本的输出	216
8.3.1	文本的输出函数	216
8.3.2	文本的对齐方式	218
8.3.3	文本输出颜色的设置	218
8.4	小结	221
第 9 章	使用位图与位操作	222
9.1	位图	222
9.1.1	什么是位图	222
9.1.2	位图的颜色深度	222
9.1.3	设备相关位图 DDB 和设备无关位图 DIB	223
9.1.4	CBitmap 类	224
9.2	位操作	229
9.2.1	光栅位操作 ROP	229
9.2.2	位块传送	229
9.2.3	光栅操作演示程序 MyROP	231
9.3	显示图标	237
9.3.1	装载图标	237
9.3.2	显示图标	239
9.4	设置光标	239
9.4.1	装载光标	239
9.4.2	设置和隐藏光标	241
9.5	MyBit 应用程序	241
9.6	小结	244
第 10 章	打印和打印预览	246
10.1	基本打印功能	246
10.1.1	标准打印对话框	247
10.1.2	CPrintInfo 类	249
10.1.3	打印作业	250
10.1.4	打印预览	252
10.2	高级打印功能设计	253
10.2.1	打印尺寸	253
10.2.2	打印分页	256
10.2.3	页眉和页脚	257
10.3	小结	261

第 11 章 多媒体编程	262
11.1 播放 wav 文件	262
11.1.1 wav 文件的技术参数	262
11.1.2 播放 wav 文件	262
11.1.3 用 MCI 函数播放声音文件	268
11.2 动画	274
11.2.1 创建 moon 应用程序框架	275
11.2.2 创建动画的资源文件	275
11.2.3 应用程序主窗口的可视化实现	277
11.2.4 编写动画的应用程序	277
11.2.5 编写 Move 动画程序	286
11.3 小结	287
第 12 章 访问数据库	289
12.1 理解数据库的概念	289
12.1.1 什么是数据库	289
12.1.2 常用数据库简介	289
12.1.3 Visual C++ 的数据库接口	290
12.2 MFC 和 ODBC、DAO	290
12.2.1 什么是 ODBC	290
12.2.2 什么是 DAO	291
12.2.3 ODBC、DAO 和 SQL 查询语言	292
12.2.4 MFC 对 ODBC API 和 DAO 的封装	293
12.2.5 创建数据库应用程序	294
12.3 MFC 的数据库类 (ODBC)	295
12.3.1 CDatabase 类	295
12.3.2 CRecordset 类	299
12.3.3 CRecordView 类	310
12.3.4 应用 ODBC 编程	312
12.4 MFC 的数据库类 (DAO)	317
12.4.1 DAO 数据库类和 ODBC 数据库类的区别	318
12.4.2 CDaoWorkspace 类	318
12.4.3 CDaoDatabase 类	321
12.4.4 CDaoTabledef 类	324
12.4.5 CDaoQuerydef 类	326
12.4.6 CDaoRecordset 类	328
12.4.7 CDaoRecordView 类	332
12.4.8 CDaoException 类	333
12.5 DAO 数据库类应用程序 MyDao	334

12.5.1	用 Microsoft Access 创建 DAO 数据库文件	335
12.5.2	创建应用程序框架	335
12.5.3	在记录视中添加数据库控件	335
12.5.4	将控件变量与数据库变量链接	336
12.5.5	实现排序、查询功能	337
12.5.6	在数据库中添加、修改和删除记录	341
12.6	小结	346
第 13 章	创建用户定义的动态链接库	347
13.1	什么是动态链接库	347
13.2	动态链接库的链接方式	347
13.3	建立一个动态链接库	349
13.3.1	建立框架结构	349
13.3.2	动态链接库库函数导出方式	352
13.3.3	添加源代码	353
13.3.4	源程序清单	354
13.4	创建一个应用程序	356
13.5	动态链接库的调试	361
13.5.1	调试一	362
13.5.2	调试二	363
13.6	小结	363
第 14 章	ActiveX 控件	364
14.1	创建 ActiveX 控件	364
14.1.1	创建 MyControl 控件的项目文件	364
14.1.2	测试 MyControl 控件	366
14.2	自定义 ActiveX 控件	367
14.2.1	修改 MyControl 控件的显示	367
14.2.2	给控件添加库存属性	369
14.2.3	库存属性的功能化	371
14.2.4	给 MyControl 控件增加属性页	373
14.2.5	给控件增加自定义属性	376
14.2.6	自定义属性的初始化	378
14.2.7	给控件增加事件	383
14.2.8	在应用程序中使用控件的事件	385
14.3	小结	387
第 15 章	Internet 编程	388
15.1	通信的概念	388
15.1.1	网络	388
15.1.2	局域网和广域网	388

15.1.3	客户机/服务器系统	389
15.1.4	调制解调器	389
15.1.5	TCP/IP 协议	389
15.2	Socket 程序设计	390
15.2.1	套接字的种类	390
15.2.2	套接字的编程要点及过程	391
15.2.3	服务器进程创建套接字	393
15.2.4	客户进程创建套接字	396
15.3	CSocket 编程模型	399
15.3.1	服务器端的 CSocket 程序	400
15.3.2	创建 SockSer 程序	401
15.4	MAPI 编程	409
15.4.1	MFC 中提供的 MAPI 支持	409
15.4.2	创建可以发送邮件的功能的应用程序	410
15.5	小结	410
第 16 章	多线程	411
16.1	启动一个新线程	411
16.1.1	用户界面线程	411
16.1.2	工人线程	412
16.2	终止线程	413
16.2.1	线程的正常终止	413
16.2.2	线程的非正常终止	414
16.3	管理新线程	414
16.3.1	CWinThread 类	415
16.3.2	暂停和恢复线程	415
16.3.3	设置线程的优先级	416
16.3.4	可操作线程的全局函数	416
16.4	同步化线程	416
16.4.1	同步化线程的原因	417
16.4.2	两个重要的 Win32 函数	418
16.4.3	同步对象的类型	419
16.5	多线程 MyThread 程序	422
16.5.1	生成 MyThread 程序框架	422
16.5.2	往应用程序里添加代码	423
16.6	小结	429
附录 A	从 C 到 C++	430
A.1	类与对象	430
A.1.1	类的定义	430

A.1.2	对象声明	431
A.1.3	类成员的访问	432
A.2	类的成员	432
A.2.1	构造函数	432
A.2.2	析构函数	435
A.2.3	静态成员变量和静态成员函数	436
A.2.4	内联成员函数	438
A.3	友元	439
A.3.1	友元函数	439
A.3.2	友元运算符函数	441
A.3.3	友元类	446
A.4	类的继承	447
A.4.1	派生类和基类	448
A.4.2	派生类的构造函数	451
A.4.3	多重继承	452
A.5	多态性	454
A.5.1	虚拟函数	454
A.5.2	虚拟函数的应用	458
A.5.3	纯虚函数和抽象基类	460
A.5.4	虚拟基类	462
A.6	小结	467
附录 B	常用类库速查	468
B.1	根类	468
B.2	应用程序体系结构类	468
B.2.1	应用程序基类	468
B.2.2	与线程相关的类	468
B.3	可视对象类	469
B.3.1	窗口类	469
B.3.2	视图类	469
B.3.3	对话类	469
B.3.4	菜单类和控制类	470
B.3.5	绘图类	470
B.4	通用类	471
B.4.1	数组和列表类	471
B.4.2	文件 I/O 类	472
B.5	OLE 类	472
B.5.1	OLE 基类	472
B.5.2	OLE 容器类	472

B.5.3	OLE 服务器类	472
B.5.4	OLE 数据传输类	473
B.5.5	OLE 对话框类	473
B.5.6	其他 OLE 类	473
B.6	数据库类	474
B.7	网络类	474
B.7.1	ISAPI 类	474
B.7.2	Windows Sockets 类	474
B.7.3	Winlnet 类	474
B.8	异常类	475

第 1 章 Windows 的编程特点

1.1 Windows 程序的特点

Windows 是一个多任务、图形化用户界面的操作系统，Windows 和用户操作是图形界面接口 GUI。Windows 的多任务，使你可以同时启动几个程序，在运行时可以在几个程序中进行切换。与 DOS 相比，它有很多优点。从编程的角度上看，它有许多资源可以利用，而分段式的编程方法，事件驱动的应用程序以及消息循环都可以使你的编程大大简化。现在我们分别介绍一下 Windows 的这些特点。

1.1.1 用户图形界面

Windows 和 DOS 的一个重要区别就是它的用户图形界面，用户界面友好与否是衡量应用程序的一个重要的方面。在 DOS 环境下，设计出一个界面友好、操作简单的图形界面是非常困难的。另外，设计出的界面一般情况下因人而异，没有一个统一的风格，以致用户使用和掌握应用程序都非常困难。而且由于风格不统一，即使学会了使用一个应用软件，对使用其他的软件也没有多大的帮助。

Windows 为设计漂亮、统一和友好的用户图形界面提供了一种全新的方法。Windows 是一个用户图形界面（GUI），各种操作对象都是以图形形式显示在屏幕上。通过键盘和鼠标，用户可以在屏幕上直接操作这些对象。这样用户和程序之间的交互不再是单一的用键盘输入信息，然后在屏幕上显示这些信息，而是直接与屏幕上的对象进行对话。

所有的 Windows 的应用程序的基本外形均差不多，每一个程序占据一个窗口。窗口是屏幕上的一个矩形区域，它是由标题栏来标识的。程序中的绝大多数的功能均由菜单来执行，但是我们可以用工具栏、对话框或是其他的子窗口连通用户和程序之间的通信。

1.1.2 多任务

在 Windows 中，多任务是另一个重要的特征。在标准的 MS-DOS 环境下编写程序，一般都假定它独占计算机的所有资源，这些资源包括输入、输出设备，内存、系统显示器和 CPU 等；而在 Windows 环境下，应用程序必须和当前正在运行的其他程序共享这些资源。因此设计程序时，要考虑多任务程序的特点并共享资源，以避免耗尽资源。

多任务具有很多优点，在应用程序中共享资源可以使操作非常简单，可以在程序中任意切换，可以在不同的文档中进行拷贝。多任务也可以节省时间，便于统筹安排工作，当进行打印或是格式化磁盘这种比较费时的工作时，可以同时进行别的工作，如编程等，当然也可以玩一玩接龙等小游戏轻松一下。

现在我们来看一看在 Windows 中是如何实现多任务的。多任务就是能同时执行几个程

序的意思。所谓同时执行，并非指一般意义上的同时，而是指每一个程序都有执行的机会，操作系统可以允许这几个程序轮流执行。从一个较长的时间上看，每一个程序都在执行。但是从一个极短的时间上看，一个时刻只能有一个程序在运行。因为一般的计算机都是一个处理器，一个时刻只能有一个程序占用处理器。

多任务分抢先式多任务和非抢先式多任务。所谓抢先式多任务，就是指无论一个应用程序执行到什么地方，只要处理器认为该让另一个程序运行了，它就会强行终止原来的应用程序，然后将原来的应用程序的各种信息如寄存器的值等保存下来，然后再装入另一个程序并执行它。一般情况下，这种方式是操作系统获得控制权，操作系统进行任务的调度，即进行程序之间的切换。而非抢先方式多任务是指由应用程序获得控制权，只有当一个应用程序结束之后才允许执行另一个应用程序。

很多多任务的操作系统，如 Windows 95、UNIX、OS/2 和 Windows NT 等都采用抢先式多任务方式。在这种方式中，必须要有完善的解决系统资源争用的方法，否则将会出现应用程序之间互相冲突的现象。

1.1.3 Windows 编程的特点

1. 面向对象的编程思想

在 Windows 的编程中，一般都是采用面向对象的编程方式。这是目前通用的编程设计思想。那么什么是面向对象的编程呢？面向对象的编程方法是一种新的设计和构造软件的思维方法，该方法的目标是使程序员在计算机上解决问题的方式更加类似于人类活动。它认为世界是由一些能独立处理内部事务的实体即对象构成，而程序中的对象就是对现实世界中对象的映射。程序员的主要任务就是对这些对象进行处理。面向对象的编程方式让程序员将工作分成几个部分，这些部分称为类，每一个类对自己的行为负责。

在 Windows 中已经给我们预定义了许多的对象，如窗口、对话框、菜单、按钮这些对象。

如前所述，窗口是屏幕上的一个矩形区域，窗口通过键盘和鼠标来接受用户的输入，并将图形输出到屏幕上。

菜单是 Windows 编程必须有的主要特征，一个菜单是一组可供用户选择和查看的命令的列表，用户只要选择菜单项就可以执行相应的命令了。

对话框是一个特殊的窗口，它为用户显示更多的有关命令的信息，并能接收用户的输入信息。一个对话框可以包含一个或多个控件。控件也是一个特殊的窗口，称为子窗口控制。控件有很多种类，包括各种按钮、文本编辑框、列表框、组合框、滚动条和其他的一些在 Windows 登记了的 ActiveX 控件，这些都可以接收用户的输入和选择。

在 Windows 的编程中，这些对象都是 Windows 的资源，用户可以使用这些资源。通过简单的接口，可以知道这些对象的功能并使用它们，在本书的后面将详细讲述这些对象。而这些资源在程序中又都是共享的，即在一个程序中使用了—个控件，在其他的程序也可以使用这个控件。通过 ActiveX 的编程，可以把我们的应用程序编程按控件的方式注册到 Windows 中，以实现这些程序的共享。

2. 消息驱动的程序结构

在执行一个 DOS 应用程序时，该程序必须取得 CPU 的控制权，它会要求 MS-DOS 执行某些功能，例如通过系统调用来不时地读取文件等。但是在 Windows 的应用程序中，情况却完全相反：Windows 掌握着控制权，例如，在处理程序中，当移动鼠标、打开窗口或执行其他动作时，在大部分时间内，都是 Windows 保留着控制权。只有当你要求 Windows 做一些事，但 Windows 不知道如何去做时，比如重新格式化一个文件，这时 Windows 才将控制权交给应用程序。

Windows 消息提供了应用程序与应用程序之间和应用程序与 Windows 之间进行通信的手段，几乎每个 Windows 程序所做的事情都是为了响应发送到窗口函数的某条消息。在多数的应用程序中，很大一部分代码的内容都是处理这些消息的。

Windows 的消息由三部分组成：消息号、字参数和长参数。消息号是预先定义的消息名标志符，字参数和长参数是与消息号相关的值。例如，当光标在窗口用户区时，我们敲击键盘时，由 Windows 向该窗口发送一个 WM_CHAR 的消息，这个宏定义在 Windows.h 中定义。窗口接到该消息后，将进行相应的处理。Windows 应用程序创建的每一个窗口都有一个窗口函数，这个窗口函数，用以处理发送到该窗口的消息。窗口函数由 Windows 采用消息驱动的方式直接调用，而不是由应用程序显式地调用。窗口函数处理完消息之后又将控制权返回给 Windows。

Windows 中有一个系统消息队列，简称消息队列。当开始执行一个 Windows 的应用程序时，Windows 为该应用程序创建一个消息队列，这个队列将存放该程序中的所有可能创建的各种窗口的所有的消息。在程序中有一段称为“消息循环”的代码，用来从消息队列中检索这些消息并把它们发送到相应的窗口中去，有些消息不必放在消息队列中，而是直接送到窗口函数中去。系统的消息队列和应用程序中的消息队列以及窗口函数之间的关系如图 1-1 所示。

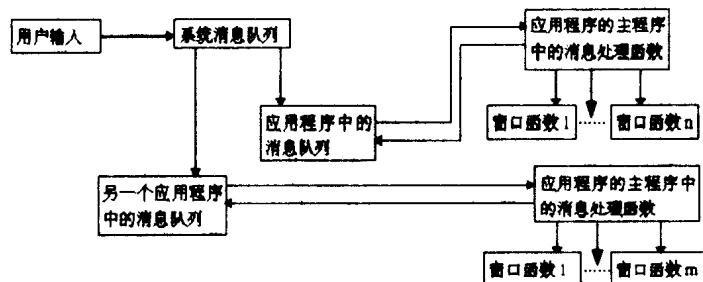


图 1-1 Windows 程序中的消息处理

Windows 的应用程序接收用户输入的方式也不同于 DOS 环境下的用户输入。在 DOS 环境下，程序通过一个函数 `getch()` 来接收键盘的输入，该函数在一个用户按下一个键后把相应的字符返回给应用程序，而在 Windows 中，Windows 将接收所有的键盘和鼠标等外设的输入，并把它们放入消息循环中。在应用程序中获取的输入只不过是简单地在消息队列中读取一个输入的消息。