Windows3.1 编程入门系列讲座(中)

第二讲　windows 3.1 应用程序的编程模式

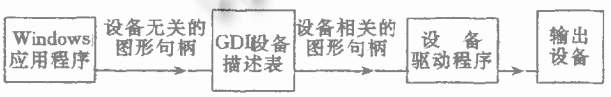
李晓华　(云南省军区自动化站)

一、windows 3.1 环境下应用程序的主要特点

1.**直观的用户图形界面**。windows 3.1 环境下应用程序具有直观、友好的用户界面,它是由图形接口来实现的,图形用户接口主要包括窗口、菜单、会话框等组成。

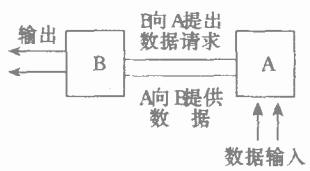
2.**队列式输入** windows 环境下应用程序将输入放入队列中,当 windows 应用程序准备好接收输入时,则从队列中取出这一输入。

3.**设备无关的图形输出**。windows 为用户提供丰富的与设备类型无关的图形,是通过以下方式实现与输出类型无关的。



4.**资源共享**。windows 环境下应用程序具有共享内存和共享 CRT 的功能。

5.**动态数据交换**(DDE)。它是 window 环境下同时运行的应用程序之间的数据交换方式之一。如:



A,B 是在 windows 环境下同时运行的两个程序。

A:实时接收数据的动态处理程序。

B:数据统计并输出。

6.**对象、连接的嵌入**(OLE)。是 windows 环境下应用程序之间进行数据交换的另一种方式。(见图 1)

7.**动态链接库**(DLL)。支持 windows 下应用程序共享数据和程序代码的一种重要技术。

8.**多任务**。

二、windows 环境下有关概念

在了解 windows 环境下应用程序编程模式之前,让我们先对 windows 环境下编程有关的基本概念作一介绍。这些概念对于初涉及到 windows 编程人员来说,是必须要理解和掌握的。只有对这些基本概念有了深刻的理解,才能对 windows 环境下应用程序的编程模式有所帮助。

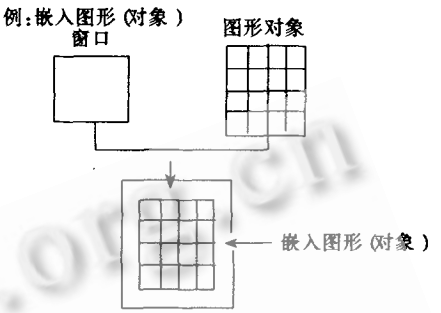


图 1

1.**对象**。是客观存在的一个实体。它具有广泛的含义,声音、图象、数据、一段文字都可以称为对象。一段应用程序也可以称之为对象,应用程序中的窗口、菜单、对话框、图象、图标等,都可统称为对象。

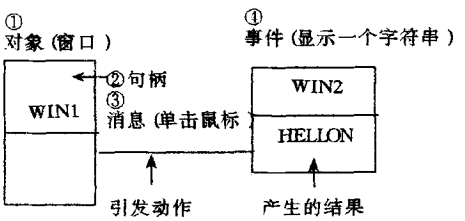
2.**面向对象的编程**。对对象的某种操作,在应用程序中这些操作可以用一段具体的代码来实现,面向对象的编程打破了传统对数据进行编程的模式,大大地扩展了编程的范围。

3.**句柄**。用户识别对象而为对象赋与的唯一名称。给一个对象赋与句柄后,对该对象的调用就可通过句柄来实现。

4.**消息**。通过某种方式传递给应用程序的信息。消息和 DOS 应用程序的输入有一定的相似之处,它们的作用都在于引发应用程序做某种动作。windows 下消息所包含的信息要比 DOS 下的输入多得多。

5.**事件**。在应用程序中由消息引发的动作。

6.**实例**。在 windows 中不仅可以同时运行多个应用程序,而且可以运行一个应用程序的多个附本,其中一个附本就称为一个实例。如:



三、windows 应用程序设计模式

绝大多数 windows 应用程序使用以下手段与用户进行交互:

1·窗口

2·菜单

© 中国科学院软件研究所 http://www.c-s-a.org.cn

3·会话框

4·消息循环

有关该部分在第三讲作专题介绍。

四、windows 应用程序编程初步

1. MC 7.0 对 windows 编程常用的数据类型和结构

类　型	含　义
WORD	16 位无符号整数
LONG	32 位有符号整数
HANDLE	16 位无符号整数,用作句柄
HWND	16 位无符号整数,用作窗口句柄
LPSTR	32 位字符串地址
FARPROC	32 位函数地址

windows 常用结构

结　构	描　述
MSG	消息队列中的消息结构
WNDCLASS	字义窗口类
PANTSTRCT	定义窗口用户域的信息
RET	定义矩形

2. MC7.0v 对 windows 编程的相关函数介绍

(1)主体函数 winmain(): 它是 windows 应用程序入口,主要完成以下事件:

①初始化应用程序,完成窗口类的登记、窗口创建及其他必须的初始化工作。

②配有消息循环、处理应用程序队列中的消息。

③当消息循环到 WM－Quit 消息时,终止应用程序。

④winmain()函数格式如下:

```
int PASCAL FAR WinMain ( hInstance, hPrevInstance,
lpCmdLine, nCmdShow)
HANDLE hInstance;
HANDLE hPrevInstance;
LPSTR lpCmdline;
int nCmdShow;
{
消息循环;
}
```

其中:

HANDLE hInstance 应用程序的实例句柄(当前实例)

HANDLE hPrevInstance 另一个实例句柄,如果没有别的事例,则为 NULL(前一个实例)

LPSTR lpCmdline 一个指向以空字符结束的命令行的长整型指针(命令行) int nCmdShow 它表明应用程序的窗口是以窗口还是以图标方式显示。

(2)和消息处理相关函数: 应用程序运行的过程就是消息的接收和处理的过程。主要包括消息接收、消息的翻译、消息的分发等。

```
long FAR PASCAL WndProc(hWnd, messagee, wParam, lParam)
HWND hWnd; 窗口句柄
unsigned message; 消息类型
WORD wParam; 附加信息
LONG lParam; 附加信息
{ PAINTSTRUCT ps;
    switch(message);
    {
    case WM-DESTROY:
        psotQuitMessage(0);
        break;
    case WM-PAINT:
        BeginPaint(hWnd, (LPPAINTSTRUCT)&ps);
        EditPaint(ps.hdc);
        EndPaint(hWnd, (LPPAINTSTRUCT)&ps);
        break;
    ..........
    default:
        return DEfWindowsProc ( hWnd, message, wParam,
lParam);
        break;
    }
}
```

(3)和窗口操作相关的函数:窗口的创建、显示、关闭、启动等。有关这部分见第三讲。

(4)和菜单操作相关的函数:

CreateMenu———————— 创建

GetMenu————————————获得句柄

GetmenuItid————————————菜单项的标识符

GetMenuItemCount———————— 确定菜单项目

GetMenuState———————— 检取状态标志

Ismenu————————————判断是否为菜单的句柄

(5)和会话相关的函数:

GreatDialog————————————创建

© 中国科学院软件研究所 http://www.c-s-a.org.cn

GetDigItemText———————— 获取信息、标题和文字

(6)和图标相关的函数：建立、打开，在固定位置显示、拷贝字符等。

GreateIcon ———————— 创建

OpenIcon ———————— 打开

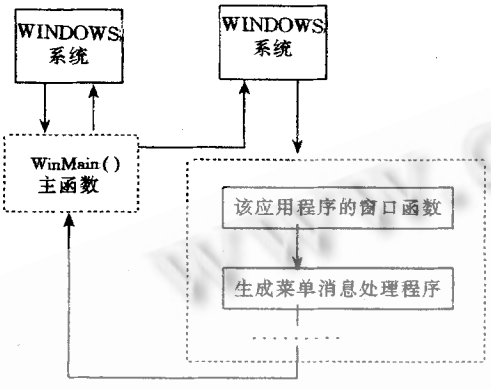
DrawIcon———————————— 画一个图符

CopyIcon———————————— 拷贝

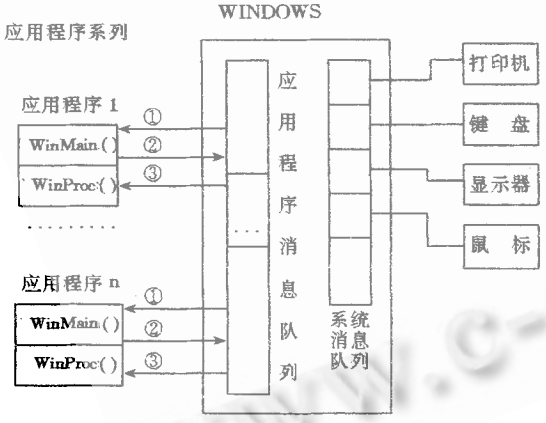
3. windows 应用程序的基本框架结构

(1)windows 应用程序的基本框架结构

①WINDOWS 应用程序总体流程图：



②WINDOWS 应用程序消息流程图：



WINDOWS 消息流程

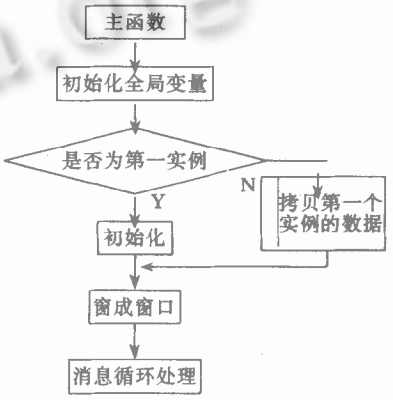
windows 是基于消息的事件驱动式程序设计的。所谓事件驱动是指驱动的程序不由事先编好的顺序来控制，而由事件的发生来控制。并且是以消息作为基础来实现的。而在其中每个事件总对应一个消息。因此 windows 的程序设计就是围绕着消息的产生、消息的处理而进行的。windows 可以同时为几个应用程序接收、发送消息。如图示，WINDOWS 收集系统队列中以消息形式的输入，然后把它们拷入到相对应的应用程序队列中。最后每个应用程序的消息循环检索自己队列中的消息，通过 WINDOWS 将其发送到相应的窗口函数。

(2)具有多道任务处理：windows 可以同时处理多个应用程序，它是以这样的方式实现的：

①系统消息队列及应用程序消息队列。windows 为每个运行着的程序维持一个消息队列，同时它监视着所有的输入设备。通常它将捕获的消息先放入到称之为"系统消息的队列"中，然后分析判断，并发送到相应的"应用消息队列"中。

②每个应用程序都有一个消息循环。windows 应用程序都对应一个消息循环，负责从各自的消息队列中检索消息，并将其发送给相应的过程来处理。

(3)winmain( )函数的框架流程：



4. WINDOWS 库

和 C 运行库一样，WINDOWS 函数已定义在一些库中. 只是为最大限度地减少应用程序的代码量，与其用户应用程序链接是在系统装载应用程序时进行的，即 DDL。

WINDOWS 主要有以下几个库：

·User 提供窗口管理

·Kernel 提供系统服务

·GDI 提供图形接口

(下期续)

·投稿须知·

·内容开门见山，直接进入主题；

·文稿尽量用打印稿，行距不宜过小，插图必须描绘清晰；

·程序不宜过长，若超过 150 行请指出重要段落及可删略部分，一律上机调试通过，并注明软、硬件运行环境；

·参考文献只指明主要 2～3 篇。

© 中国科学院软件研究所 http://www.c-s-a.org.cn