[C语言Windows程序设计->第三天->属于自己的窗口](http://www.cnblogs.com/mr-wid/archive/2012/10/08/2715673.html)

2012-10-08 18:43 by wid, 15304 阅读, 7 评论, [收藏](http://www.cnblogs.com/mr-wid/archive/2012/10/08/2715673.html), [编辑](http://i.cnblogs.com/EditPosts.aspx?postid=2715673)

**创建属于自己的窗口**

**==================**

**·创建窗口前的准备**

在创建窗口前我们先来熟悉几个名词, 这些名词现在可以暂时不去透彻的进行理解, 只需要印象中知道有这么回事就行。

**1>. 窗口**

　　"窗口"这个我们都已经十分熟悉了, 在Windows中, 一个应用程序窗口上一般会在标题栏上显示程序的名称, 紧挨着标题栏的菜单栏, 或许还会有状态栏、滚动条等其他"装饰品"。

**2>. 控件**

　　在一些应用程序中, 我们经常可以在程序的界面上(窗口)看到一些按钮(Push Button)、文本框(Text Box)、列表框(List Box)、滚动条(Scroll Bar)等, 这些对象通常被称为控件, 在《Windows程序设计》一书中, 还被称为"子窗口"、"控件窗口"或"子窗口控件"。

**3>. 窗口类**

　　在建立一个窗口前, 我们必须首先注册一个"窗口类"(Windows Class), 接触过面向对象的朋友应该会首先想到面向对象当中的"类", 但是, 这里"窗口类"中的"类"并不是指面向对象当中的那个"类"。

　　在这里我们可以把"窗口类"理解为一个结构体, 结构体的成员就是窗口的一些属性, 例如窗口的标题是什么、窗口使用什么样的小图标以及窗口的风格之类的属性, 一个窗口就是一个结构体的对象, 结构体成员的属性决定着窗口的属性。

**4>. 消息循环**

　　在Windows程序设计中, 消息循环是个不得不提的概念, Windows操作系统是以消息驱动的, 消息队列是指在一个应用程序运行时, Windows操作系统会为该应用程序建立一个"消息队列", 这个消息队列用来存放该程序可能创建的各种窗口的消息, 当用户对应用程序进行操作时, 例如点击一个按钮、调整下窗口的大小等, 此时Windows会立即把这一消息告诉应用程序, 使应用程序能作出相应的动作。

　　笔者觉得有比较强调一下上一段中的最后一句"此时Windows会立即把这一消息告诉应用程序, 使应用程序能作出相应的动作。", 我们在创建一个应用程序时, 不用想着什么时候才能从用户那得到指令, 因为操作系统会即时告诉我们用户此时是否对程序进行了操作, 对于没有接触过Windows编程的朋友们这点可能有点难以理解, 简单来说就是像如何获取一个按钮是否被单击, 或者如何获取用户此时是否在调整窗口大小之类的代码Windows已经帮我们完成了, 我们只需要等待着Windows给我们发消息就行, 如何判断消息类型以及处理这些消息, 在代码上通常我们用while配合一个巨大的switch来完成,。

**5>. 窗口的过程函数**

　　当一个窗口建立之后, 就可以从Windows那里不断的接收到Windows发来的消息, 接收到消息后我们就需要一个函数来处理这些消息, 可以简单的理解为, 这个用来处理接收到的消息的函数就称为窗口过程函数或者回调函数。

下面我们来尝试着看一个创建一个基本窗口的代码。

**·尝试创建自己的窗口**

代码如下:

按 Ctrl+C 复制代码



按 Ctrl+C 复制代码

--------------------

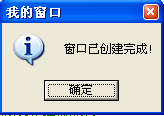


--------------------

　　先简单介绍这段代码, 在主函数WinMain中的窗口类对象wndclass我们定义了窗口的相关属性, 尝试注册窗口类并调用CreateWindow创建窗口, 创建完成后用ShowWindow让窗口显示出来, 我们还使用了个while用来从消息队列里获取并分发消息给程序, 我们还定义了一个过程函数WndProc用来处理系统发来的消息。

　　编译运行后, 如果没有错误将首先看到一个窗口建立成功与否的对话框, 当创建窗口失败时弹出一个错误对话框并关闭程序。

　　当窗口被创建时首先Windows会发给我们一条创建完成的消息"WM\_CREATE", 在我们的消息处理函数WinProc中有switch-case语句对该消息进行了处理, 就是弹出一个窗口创建完成的对话框。

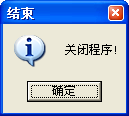


　　同样, 我们还处理了当窗口的客户区(内容部分)被改变时发来的的"WM\_PAINT"消息进行了处理, 让"Hello, 这是我创建的窗口!"始终显示在窗口中心。

"WM\_LBUTTONDOWN"消息是当用户在客户区按下鼠标左键时发来的消息, 我们作出相应的动作为弹出一个鼠标左键被按下的对话框。



当用户点击窗口的关闭按钮时, "WM\_DESTROY"消息就会发来, 我们作出的动作是弹出"关闭程序!"的对话框并且退出程序。



今天的学习先到这里, 明天我们详细学习下整个创建窗口的代码。