

PC技术辅导:窗口子类化 PDF转换可能丢失图片或格式，建议
阅读原文

https://www.100test.com/kao_ti2020/259/2021_2022_PC_E6_8A_80_E6_9C_AF_E8_BE_c98_259401.htm

理论：如果你曾经在 Windows 环境下编过程序，有时候就会发现：有一个现成的窗口，几乎有你所需要的全部功能，但还不完全一样（否则就没有必要讲这一节了）。你曾遇到过这样的处境吗，如果你需要一个具有过滤特殊字符功能的 Edit 控件。当然最直接的方法就是自己用代码来实现，但这的确是一个费时又很困难的任务，而窗口子类化就可以用来做这种事情。窗口子类化允许你接管被子类化的窗口，使你对它有绝对的控制权。举个例子了来阐明一下：例如你需要一个只接受十六进制数字输入的文本编辑框，如果使用一个简单的 Edit 控件，当用户输入十六进制以外的字符时，你既不知道也无计可施。也就是说，当用户进文本框中输入字符串 "zb q*" 时，如果除了拒绝接受整个字符串以外几乎什么也不能做，至少这显得特别不专业。重要的是，你需要具有输入检测的能力，即每当用户输入一个字符到编辑框中时要能检测这个字符。现在来解释实现细节：当用户往文本框中输入字符时，Windows 会给 Edit 控件的窗口函数发送 WM_CHAR 消息。这个窗口函数本身寄生于 Windows 中，因此不能直接修改它。但是我们可以重定向这个消息使之发送到我们自己编写的窗口处理函数。如果自定义窗口要处理这个消息那就可以处理它，如果不处理就可以把这个消息转发到它原来窗口处理函数。通过这种方式，自定义的窗口处理函数就把它自己插入到 Windows 系统和 Edit 控件之间。看下面的流程：窗口子类化之前

Windows ==> Edit 控件的窗口处理函数。子类化之后
Windows ==> 自定义的窗口处理函数 ==> Edit 控件的窗口处理函数。注意子类化并不局限于控件，可以子类化任何窗口，现在我们要把精力集中到怎样实现子类化一个窗口上。让我们想想Windows 怎样知道 Edit 控件的窗口处理函数放在什么地方。猜的？...肯定不是。原来 WNDCLASSEX 结构的成员 lpfnWndProc 指出了窗口函数地址。如果能用自己编写的窗口函数的地址来替换这个成员变量，那 Windows 不就把消息发到自定义的窗口函数了吗！我们通过调用函数 SetWindowLong 来实现这个任务，此函数的原型为：SetWindowLong PROTO hWnd:DWORD, nIndex:DWORD, dwNewLong:DWORD hWnd = 将要实施子类化的窗口的句柄 nIndex = 函数了功能索引
GWL_EXSTYLE 设置窗口的扩展风格. GWL_STYLE 设置新的窗口风格
GWL_WNDPROC 设置新的窗口处理函数地址
GWL_HINSTANCE 设置新的应用程序句柄
GWL_ID 设置新的窗口标识
GWL_USERDATA 设置一个与这个窗口相关的给用户使用的32位的数据
dwNewLong = 用来更新的数据
我们的工作还是比较简单的：写一个窗口函数用于处理发给 Edit 控件的消息。用参数 GWL_WNDPROC 调用 SetWindowLong 函数，如果调用成功那么返回值就是与调用功能相联系的一个32位的整数
在我们的程序中，返回值就是原先窗口函数的地址。我们要保存这个值以便以后使用。记住：有一些我们不处理的消息，需要把它们派遣给原来的窗口函数来处理，这就用到另外一个函数 CallWindowProc，函数原型为：
CallWindowProc PROTO lpPrevWndFunc:DWORD, hWnd:DWORD, Msg:DWORD, wParam:DWORD,

lParam:DWORD lpPrevWndFunc = 窗口原来函数的地址. 剩下的四个参数就是发给自定义函数的参数，直接把它们传给函数 CallWindowProc 就行了。100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com