<<老码识途>>

图书基本信息

书名：<<老码识途>>

13位ISBN编号：9787121173820

10位ISBN编号：7121173824

出版时间：2012-8

出版时间：电子工业出版社

作者：韩宏

页数：344

<<老码识途>>

内容概要

本书以逆向反汇编为线索，自底向上，从探索者的角度，原生态地刻画了对系统机制的学习，以及相关问题的猜测、追踪和解决过程，展现了系统级思维方式的淬炼方法。
该思维方式是架构师应具备的一种重要素质。
本书内容涉及反汇编、底层调试、链接、加载、钩子、异常处理、测试驱动开发、对象模型和机制、线程类封装、跨平台技术、插件框架、设计模式、GUI框架设计等。
书中包含不少工业级或非公开案例。
读者不仅能以底层观和调试技巧解决各种实际问题；还可掌握一套学习方法，如“猜测—实证—构建”，调构学习法。

<<老码识途>>

书籍目录

章节摘录

 插图： 那么，登记窗体句柄和处理函数的对应关系有两种方法。
①在窗体描述结构WNDCLASS中指出。
窗体千差万别，如果让系统构造一个窗体，需要给系统一份窗体图纸。
该图纸就是WNDCLASS结构体，其中有两个重要的成员变量，一个是DM7—2的第l3行的lpszClassName，标识这份图纸的名字，例中给定名字为“testwin”。
后面构造窗体时，就用这个名字告知系统我们想用哪份图纸构造，第15行CreateWindow构造窗体，其第1个参数就是“testwin”，指出了图纸的名字。
要用这份图纸自然需要将它注册给系统，所以第14行RegisterClass完成了该工作。
WNDCLASS结构体另一个成员是第l2行的lpfnWndProc，指出了窗体处理函数的地址，指示的函数就是第1～9行的函数WndProc（）。
这种方法指出窗体的处理函数，意味着所有用名为“testwin”的WNDCLASS构造出的窗体都拥有同样的处理函数，如果构造了多个窗体，如何让函数区分到底当前在处理哪个窗体的消息？
DM7—2中，第1行窗体处理函数的第1个参数hwnd指出了被处理消息属于哪个窗体。
只要在函数中用判断hwnd的不同，即可做出相应处理，这样一个窗体函数可被多个窗体共用，却又能各行其是。
②第①种方法为某类型窗体指定相同处理函数，这种方法可为某个具体窗体对象指定处理函数。
也就是说，可以用同一个WNDCLASS结构创建出的窗体拥有各自不同的处理函数。
调用函数SetWindowLong（）即可，SetWindowLong（hwnd，GWL WNDPROC，（LONG）wndProc）将消息处理函数指针wndProc设定给句柄hwnd指示的窗体。
消息处理函数怎样处理消息？
从DM7—2第1行可知，它有4个参数。
第1个已解释，是接收消息的窗体的句柄。
第2个是无符号整数，代表消息类型，如宏WM_PAINT就是重绘消息。
从winuser.h中可知，该宏声明为#defme WM_PAINT 0x000F，即说明十进制数15代表重绘消息。
一个消息可能包含相关参数，则由第3、4个参数指出，其类型为指针，但很多时候被直接当成整数。
比如，鼠标左键按下消息WM_LBUTTONDOWN中，wParam的整数值代表了此时其他键按下的状态，如MK SHIFT代表键盘Shift键是否按下。
读者可能会问，如果有2个甚至以上的参数需要表示怎么办？
系统非常节约，比如还是WM_LBUTTONDOWN消息，需要包含此时鼠标的坐标x和y，它用1Param的高2字节代表y，低2字节代表X。
如果还有更多信息呢？
因为wParam和lParam都声明为指针，那么就可以将任何结构体的指针作为wParam或1Param传递即可。
这与第5章线程函数的参数只接收一个void*类型却可传递任何信息是一样的。

<<老码识途>>

编辑推荐

《老"码"识途:从机器码到框架的系统观逆向修炼之路》包含不少工业级或非公开案例，读者不仅能以底层观和调试技巧解决各种实际问题；还可掌握一套学习方法，如“猜测—实证—构建”，调构学习法。

版权说明