

图形界面篇



日成蝶—Windows API 编程入门

七日做茧，一朝成蝶！



主讲：袁春旭

自建窗口

自建窗口

步骤说明：

步骤一：注册窗口类（RegisterClassEx）

步骤二：创建窗口（CreateWindowEx）

步骤三：显示/更新窗口（ShowWindow/UpdateWindow）

步骤四：消息循环及消息处理（GetMessage/DispatchMessage）

自建窗口-API

窗口注册

```
ATOM WINAPI RegisterClassEx(  
    const WNDCLASSEX *lpwctx  
);
```

lpwctx: WNDCLASSEX的指针。

返回值: unsigned short, 失败返回0, 成功返回窗口标识

typedef unsigned short ATOM;

自建窗口-API

```
typedef struct tagWNDCLASSEX {  
    UINT    cbSize;        //结构体大小  
    UINT    style;         //Window Class Styles  
    WNDPROC lpfnWndProc;   //消息处理函数  
    int     cbClsExtra;    //设为0  
    int     cbWndExtra;    //设为0  
    HINSTANCE hInstance;  //窗口处理句柄  
    HICON     hIcon;       //应用程序图标句柄  
    HCURSOR   hCursor;    //鼠标资源句柄  
    HBRUSH    hbrBackground; //窗口背景画刷句柄  
    LPCTSTR   lpszMenuName; //窗口菜单名称  
    LPCTSTR   lpszClassName; //注册的窗口类名称  
    HICON     hIconSm;    //窗口图标句柄  
} WNDCLASSEX, *PWNDCLASSEX;
```

自建窗口-API

```
HWND WINAPI CreateWindow(  
    LPCTSTR lpClassName, //注册的类名  
    LPCTSTR lpWindowName, //窗口名称  
    DWORD dwStyle,        //窗口风格  
    int x,                //窗口左上角横坐标  
    int y,                //窗口左上角纵坐标  
    int nWidth,           //窗口宽度  
    int nHeight,          //窗口高度  
    HWND hWndParent,      //父窗口句柄  
    HMENU hMenu,          //菜单句柄  
    HINSTANCE hInstance,  //实例句柄  
    LPVOID lpParam        //额外参数  
);
```

自建窗口-API

```
HWND WINAPI CreateWindowEx(  
    DWORD    dwExStyle,        //窗口样式扩展  
    LPCTSTR  lpClassName,      //注册的类名  
    LPCTSTR  lpWindowName,     //窗口名称  
    DWORD    dwStyle,          //窗口风格  
    int      x,                //窗口左上角横坐标  
    int      y,                //窗口左上角纵坐标  
    int      nWidth,           //窗口宽度  
    int      nHeight,          //窗口高度  
    HWND     hWndParent,       //父窗口句柄  
    HMENU     hMenu,           //菜单句柄  
    HINSTANCE hInstance,       //实例句柄  
    LPVOID    lpParam          //额外参数  
);
```


自建窗口-API

显示窗口

```
BOOL ShowWindow(  
    HWND hWnd,  
    int nCmdShow  
);
```

hWnd：窗口句柄。

nCmdShow：控制窗口显示行为。

返回值：true显示窗口，false隐藏窗口

自建窗口-API

nCmdShow取值	含义
SW_HIDE (0)	隐藏窗口
SW_MAXIMIZE (3)	窗口最大化
SW_MINIMIZE (6)	窗口最小化
SW_SHOWNORMAL (1)	窗口正常化
SW_SHOW (5)	显示并激活窗口

自建窗口-API

更新窗口

```
BOOL UpdateWindow(  
    HWND hWnd  
);
```

hWnd：窗口句柄。

返回值：失败返回false，成功返回true

自建窗口-API

消息获取

```
BOOL GetMessage(  
    LPMSG lpMsg, //输出参数, 取出消息  
    HWND hWnd, //窗口句柄  
    UINT wMsgFilterMin, //消息过滤器最小值  
    UINT wMsgFilterMax //消息过滤器最大值  
);
```

返回值：如果取出的是WM_QUIT消息，返回0.

自建窗口-API

消息分发

```
LRESULT DispatchMessage(  
    const MSG *lpmsg  
);
```

lpmsg：消息结构体指针。

自建窗口-API

消息结构

```
typedef struct tagMSG {  
    HWND hwnd;      //接收消息的对象句柄  
    //消息名称定义在WinUser.h文件中  
    UINT message; //消息描述, 应用程序用低位, 高位系统保留  
    WPARAM wParam; //消息的额外信息  
    LPARAM lParam; //消息的额外信息  
    DWORD time;    //提交消息的时间  
    POINT pt;      //提交消息时鼠标的位置  
} MSG;
```

伪代码

伪代码

```
int APIENTRY WinMain(HINSTANCE hInstance,
                    HINSTANCE hPrevInstance,
                    LPTSTR lpCmdLine,
                    int nCmdShow)
{
    RegisterClassEx(...); //注册类, 并设定消息处理函数
    CreateWindowEx(...); //创建窗口
    ShowWindow(...); //显示窗口
    UpdateWindow(...); //更新窗口
    while (GetMessage(...))
    {
        DispatchMessage(...); //消息分发
    }
}
```


伪代码

```
LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)
{
    switch (message)
    {
        case WM_COMMAND: break;
        case WM_PAINT: break;
        case WM_DESTROY:
            PostQuitMessage(0);
            break;
        default:
            return DefWindowProc(hWnd, message, wParam, lParam);
    }
}
```

编码实战



Thank You !