

# 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

# 消息机制

# 消息机制-关键问题

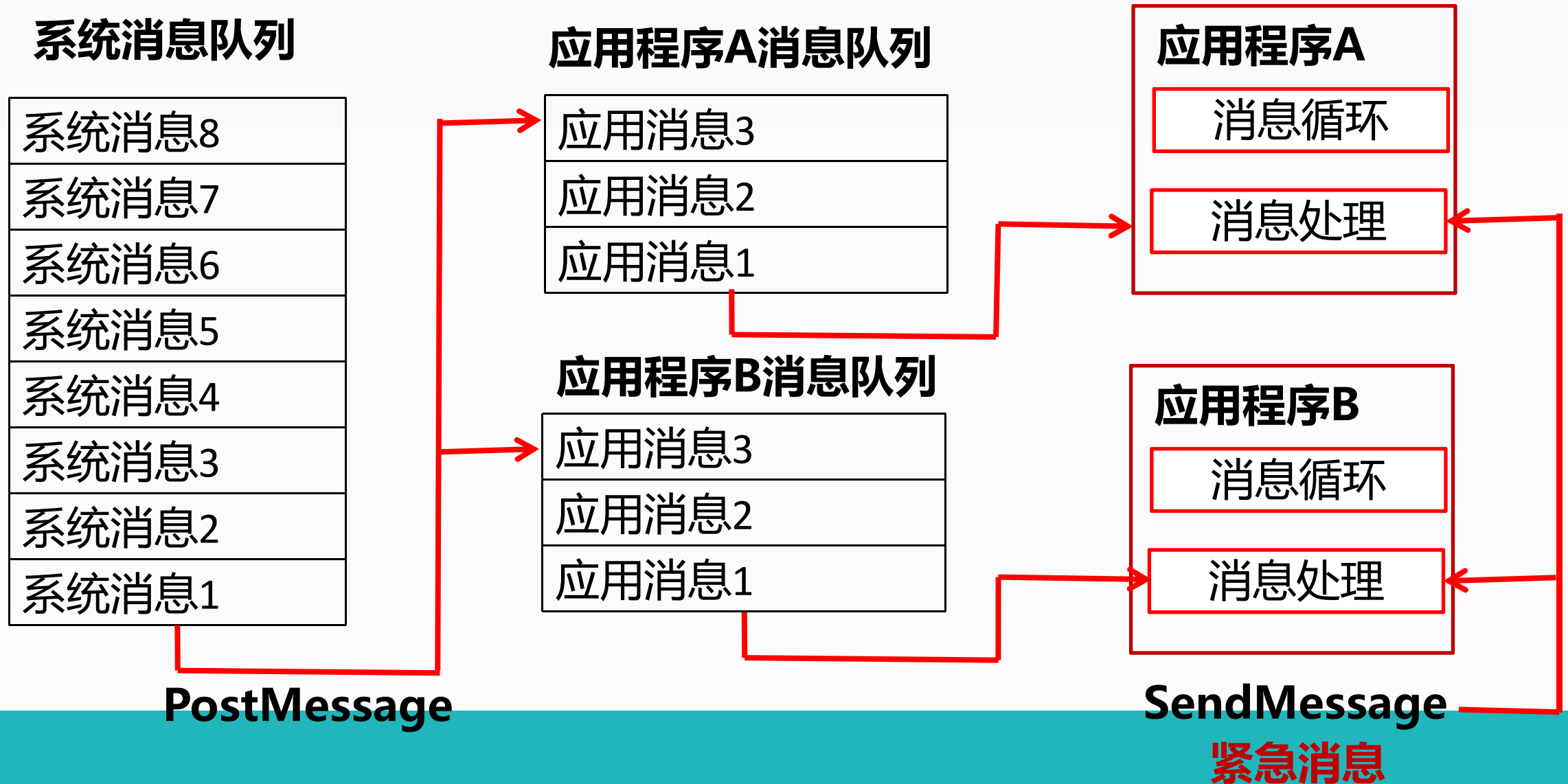
谁产生消息：系统和应用程序

消息传递方式：队列消息+非队列消息

几个队列：系统消息队列+线程消息队列（GUI）

消息类型：WM(窗口消息)、BM(按钮消息)、CB(组合框消息)

# 消息机制-传递流程



# 消息机制-发送消息

```
BOOL PostMessage(  
    HWND hWnd,  
    UINT Msg,  
    WPARAM wParam,  
    LPARAM lParam  
);
```

hWnd：窗口句柄。

Msg：消息ID。

wParam、lParam：附加参数

# 消息机制-发送消息

```
LRESULT SendMessage(  
    HWND hWnd,  
    UINT Msg,  
    WPARAM wParam,  
    LPARAM lParam  
);
```

hWnd：窗口句柄。

Msg：消息ID。

wParam、lParam：附加参数

# 消息机制-获取消息

```
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.



# 消息机制-获取消息

```
BOOL PeekMessage(  
    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  
    UINT wRemoveMsg //是否移除消息  
);
```

PM\_NOREMOVE: 不移除消息。

PM\_REMOVE: 移除消息。

# 消息机制-派发消息

```
LONG DispatchMessage(  
    CONST MSG *lpmsg  
);
```

lpmsg：消息指针。

## 默认处理API

# 消息机制-消息处理

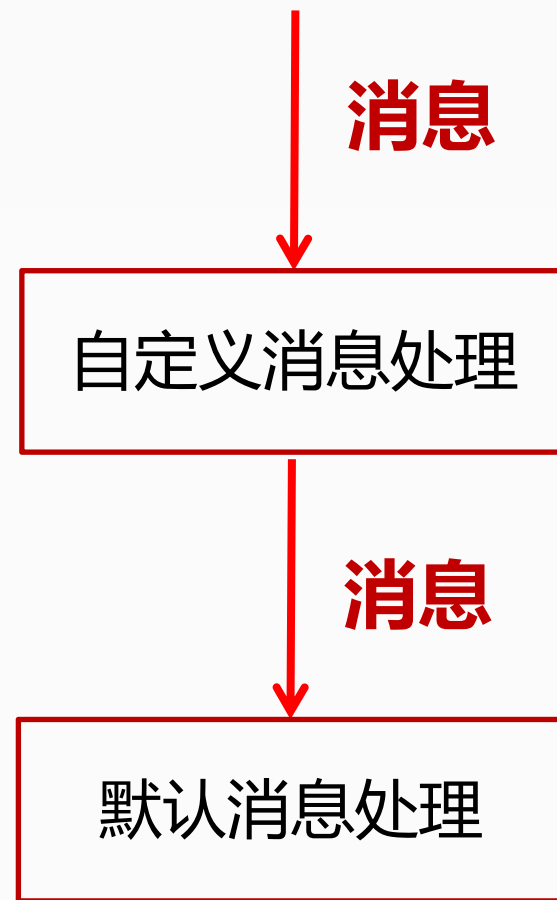
```
LRESULT DefWindowProc(  
    HWND    hWnd,  
    UINT     Msg,  
    WPARAM  wParam,  
    LPARAM  lParam  
);
```

hWnd: WNDCLASSEX的指针。

Msg: 消息ID。

wParam、lParam: 消息参数

返回值: 执行结果



# 消息机制-处理代码

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

# 消息机制-等待消息

```
BOOL WaitMessage( void );
```

功能说明：当一个线程的消息队列中没有消息存在时，WaitMessage函数会使该线程中断并处于等待状态，同时把控制权交给其它线程，直到被中断的那个线程的消息队列中有了新的消息为止。

# 消息机制-键盘消息

```
BOOL TranslateMessage(CONST MSG*lpMsg);
```



关键概念：虚拟键VK\_

lpMsg：指向MSG结构的指针。

返回值：消息转换失败返回false。

# 编码实战



# Thank You !