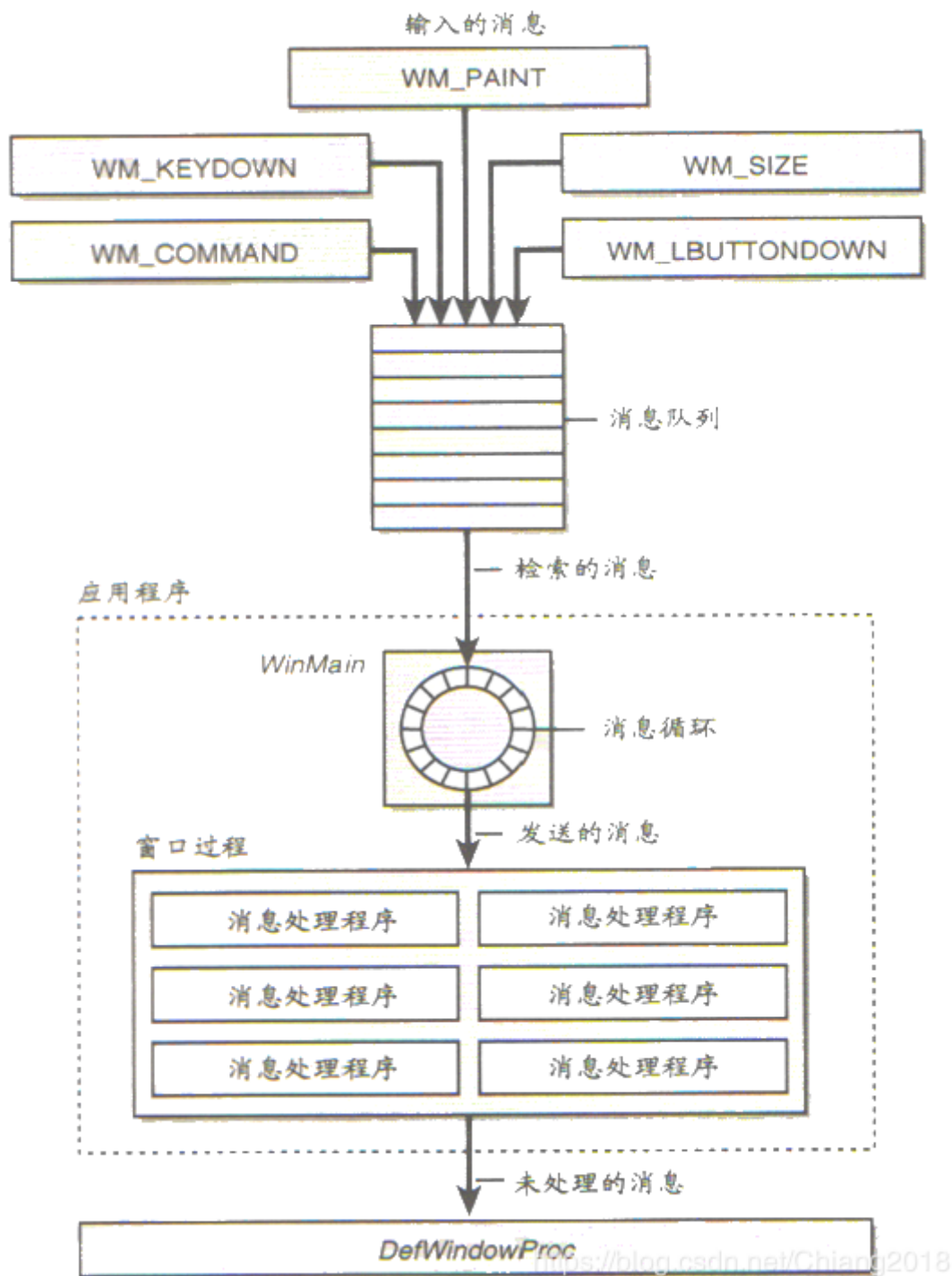


1.Windows编程模型

1. 事件驱动编程模型
2. 一切都是窗口：窗口和句柄
3. 消息和消息队列



```
#include<windows.h>
//窗口回调函数
LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);
//HINSTANCE----实例句柄
//hInstance 当前实例
//hPrevInstance 前一个实例已经淘汰
//nCmdShow控制窗口如何显示
```

```

int WINAPI WinMain(HINSTANCE hInstance ,HINSTANCE hPrevInstance,LPSTR
lpCmdLine,int nCmdShow) {
    HWND hwnd;//窗口句柄
    MSG msg;//消息
    WNDCLASS wc;//窗口类
    //1.设计一个窗口类
    wc.style = 0;//默认样式
    wc.lpfnWndProc = (WNDPROC)WndProc;//窗口过程
    wc.cbClsExtra = 0;//窗口类额外数据
    wc.cbWndExtra = 0;//窗口额外数据
    wc.hInstance = hInstance;//哪个实例的---当前实例
    wc.hIcon = LoadIcon(NULL,IDI_WINLOGO);//左上角图标
    wc.hCursor = LoadCursor(NULL, IDC_ARROW);//鼠标
    wc.hbrBackground = (HBRUSH)(COLOR_WINDOW + 1);//背景色,HBRUSH是画刷
    wc.lpszMenuName = NULL;//菜单
    wc.lpszClassName = TEXT("MyWndClass");//类名字
    //2.注册窗口
    RegisterClassW(&wc);
    //3.创建窗口
    hwnd=CreateWindow(TEXT("MyWndClass"),TEXT("Delort's windows SDK
Application"),WS_OVERLAPPEDWINDOW,
        CW_USEDEFAULT,CW_USEDEFAULT, CW_USEDEFAULT,CW_USEDEFAULT,
        NULL,NULL,hInstance,NULL);
    //类名 窗口标题 窗口样式 x,y位置 高宽 父窗口 窗口菜单句柄 当前实例

    //4.显示和更新窗口
    ShowWindow(hwnd,nCmdShow);
    UpdateWindow(hwnd);

    //5.消息循环
    while (GetMessage(&msg, NULL, 0, 0)) {
        TranslateMessage(&msg);//翻译消息,主要翻译键盘扫描码
        DispatchMessage(&msg);//转发到窗口过程
    }
    return msg.wParam;
}

LRESULT CALLBACK WndProc(HWND hwnd, UINT message, WPARAM wParam, LPARAM lParam)
{
    PAINTSTRUCT ps;//绘制结构
    HDC hdc;//DC句柄,设备上下文,可以理解为窗口内部可以显示的设备
    //对各种消息进行处理
    RECT rect;
    switch (message) {
    case WM_SIZE:
        //改变窗口大小重画
        return 0;
    case WM_LBUTTONDOWN:
        //MessageBox(hwnd,TEXT("Mouse Click"),TEXT("消息"),MB_OK);//hwnd当前窗口
        //PostQuitMessage(0);
        return 0;//消息处理完一定要return否则会转到DefWindowsProc
    case WM_PAINT://绘制
        hdc = BeginPaint(hwnd, &ps);
        GetClientRect(hwnd,&rect);
        Ellipse(hdc,0,0,200,100);//椭圆
        DrawText(hdc,TEXT("Hello windows! Hello
Delort!"),-1,&rect,DT_SINGLELINE|DT_CENTER|DT_VCENTER);

```

```
        EndPaint(hwnd, &ps);  
        return 0;  
    case WM_DESTROY: //销毁窗口(关闭)  
        PostQuitMessage(0);  
        return 0;  
    }  
    return DefWindowProc(hwnd, message, wParam, lParam); //转到默认窗口  
}
```