

## 实验一 基本窗口创建

### 一、实验目的：

- 1 掌握 Windows 编程基础知识：窗口、事件驱动及消息
- 2 掌握 Windows 应用程序的组成和基本结构

### 二、实验要求：

- 1 预习本次实验的实验要求和任务
- 2 熟悉 VC++集成开发环境并独立完成本次实验

### 三、实验环境：

- 1 软件：VC++ 6.0 或 Visual Studio

### 四、实验内容：

- 1 熟悉 VC++集成开发环境，创建一个 VC 工程，熟悉工程的工作空间
- 2 创建 cpp 文件，编写基本 Windows 应用程序的两个组成部分：WinMain 函数以及窗口函数
- 3 完成 WinMain 函数中的函数说明、窗口类的定义、窗口类的注册、窗口的创建、窗口的显示以及消息循环
- 4 完成窗口函数中的基本消息处理过程

### 五、实验过程及结果描述

- 1 在 VC++ 6.0 环境下，选择“新建”，再选择“新建”下的“工程”中的“Win32 Application”，选择要创建的工程所在文件夹，输入要创建的工程文件名
- 2 创建成功后，继续选择“新建”下的“文件”中的“中的“C++ Source File”，输入本工程的 cpp 文件名。
- 3 在本工程的 cpp 文件中输入以下代码：

WinMain 函数代码如下：

```
LRESULT CALLBACK WinSunProc(  
    HWND hwnd,          // handle to window  
    UINT uMsg,           // message identifier  
    WPARAM wParam,       // first message parameter  
    LPARAM lParam        // second message parameter  
);  
  
int WINAPI WinMain(  
    HINSTANCE hInstance,    // handle to current instance  
    HINSTANCE hPrevInstance, // handle to previous instance
```

```

LPSTR lpCmdLine,          // command line
int nCmdShow
)
{
    WNDCLASS wndcls;
    wndcls.cbClsExtra=0;
    wndcls.cbWndExtra=0;
    HBRUSH hB;
    hB=CreateSolidBrush(RGB(50, 20, 200));
    //wndcls.hbrBackground=(HBRUSH)GetStockObject(BLACK_BRUSH);
    wndcls.hbrBackground=hB;
    wndcls.hCursor=LoadCursor(NULL, IDC_CROSS);
    wndcls.hIcon=LoadIcon(NULL, IDI_ERROR);
    wndcls.hInstance=hInstance;
    wndcls.lpfnWndProc=WinSunProc;
    wndcls.lpszClassName="WHUT";
    wndcls.lpszMenuName=NULL;
    wndcls.style=CS_HREDRAW | CS_VREDRAW;
    RegisterClass(&wndcls);
    HWND hwnd;
    hwnd=CreateWindow("WHUT", "武汉理工大学计算机学院", WS_OVERLAPPEDWINDOW /*&
~WS_MINIMIZEBOX*/,
        0, 0, 600, 400, NULL, NULL, hInstance, NULL);
    ShowWindow(hwnd, SW_SHOWNORMAL/*SW_MINIMIZE*/);
    UpdateWindow(hwnd);
    MSG msg;
    while(GetMessage(&msg, NULL, 0, 0))
    {
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }
    return 0;
}

```

窗口函数代码如下:

```

LRESULT CALLBACK WinSunProc(
    HWND hwnd,          // handle to window
    UINT uMsg,          // message identifier
    WPARAM wParam,      // first message parameter
    LPARAM lParam       // second message parameter
)
{
    switch(uMsg)
    {
        //case WM_LBUTTONDOWN:
            //MessageBox(hwnd, "mouse clicked", "WHUT", 0);
    }
}

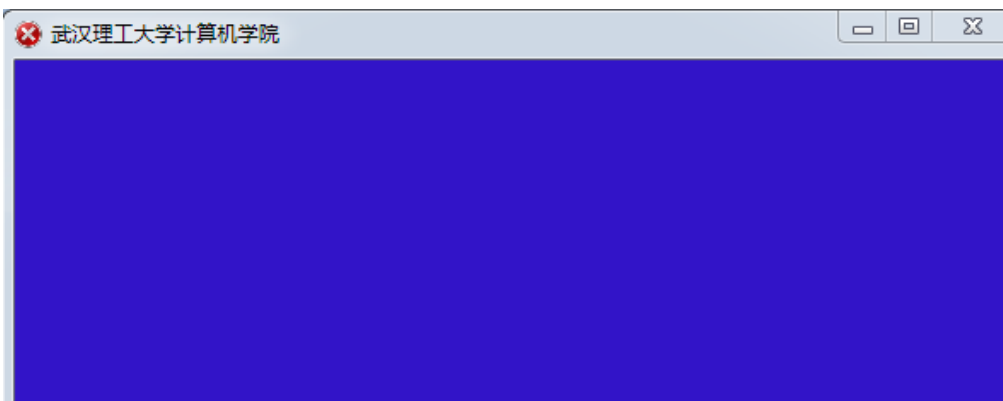
```

```

        case WM_CLOSE:
            if (IDYES==MessageBox(hwnd, "是否真的结束?", "whut", MB_YESNO))
            {
                DestroyWindow(hwnd);
            }
            break;
        case WM_DESTROY:
            PostQuitMessage(0);
            break;
        default:
            return DefWindowProc(hwnd, uMsg, wParam, lParam);
    }
    return 0;
}

```

最终的实验结果：生成的窗口截图如下：



4 在上面例子的基础上，使用定时器，每隔 1 秒，交替的用红色、蓝色和绿色来填充整个窗口的用户区。