**暨南大学本科实验报告专用纸**

课程名称 windows编程实验 成绩评定

实验项目名称 Windows编程模型 指导教师 刘晓翔

实验项目编号 实验一 实验项目类型 设计型 实验地点 三楼机房1

学生姓名 陈 伟 学号 2022100092

学院 智能科学与工程学院 系 人工智能 专业 人工智能

实验时间 2025 年6月4日上午～6月14日上午温度 28 ℃湿度

1. **实验目的**

掌握Windows编程模型与内部运行机制，掌握Windows入口函数、创建窗口、消息循环和窗口过程函数。

1. **实验内容和要求**

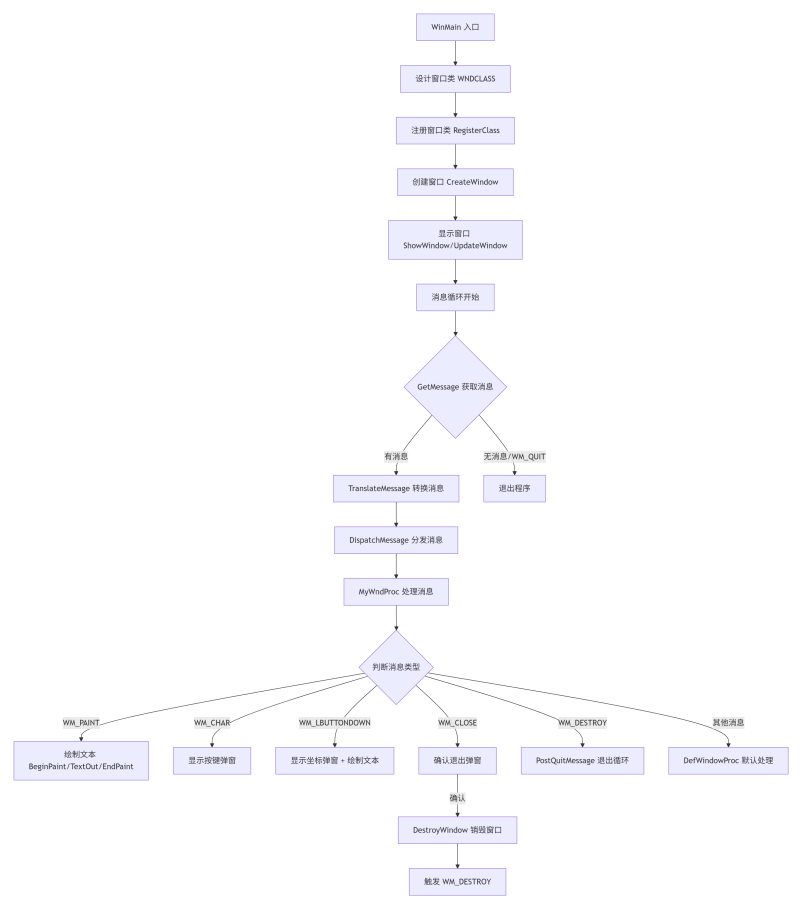
利用Visual C++6.0 Win32开发工具编写程序，通过Windows API实现一个窗口界面程序。

1. **主要仪器设备**

仪器：计算机

实验环境： Windows XP + Visual C++6.0

1. **实验原理**



1. **实验步骤与调试**

1. 入口函数 WinMain

​​作用：Windows桌面程序的入口点，类似于控制台程序的 main() 函数。

​​关键步骤：

​​设计窗口类(WNDCLASS)：

定义窗口的外观和行为（如背景色、光标、图标等）。

指定窗口过程函数 MyWndProc（用于处理消息）。

​​注册窗口类 (RegisterClass)：

将设计好的窗口类注册到操作系统中。

​​创建窗口（CreateWindow)：

根据注册的类名创建窗口实例。

设置窗口标题、样式、位置和大小等。

​​显示窗口(ShowWindow + UpdateWindow)：

显示窗口并强制立即绘制（触发 WM\_PAINT 消息）。

2. 消息循环

​​作用：持续从系统消息队列中获取消息并分发给窗口过程。

​​关键函数：

GetMessage：从消息队列中获取消息（阻塞直到有消息）。

TranslateMessage：将虚拟键消息转换为字符消息（如 WM\_KEYDOWN → WM\_CHAR）。

DispatchMessage：将消息发送到窗口过程 MyWndProc。

​​退出条件：

当收到 WM\_QUIT 消息时（由 PostQuitMessage 生成），GetMessage 返回 0，循环结束。

3. 窗口过程函数 MyWndProc

​​作用：处理发送到窗口的所有消息（事件驱动）。

​​消息处理逻辑：

​​WM\_PAINT：窗口需要重绘时触发（如首次显示、窗口被拖动）。

调用 BeginPaint 和 EndPaint 获取和释放设备上下文（DC）。

使用 TextOut 在窗口上绘制文本。

​​WM\_CHAR：用户按下字符键时触发。

显示按键的 ASCII 码（wParam）的弹窗。

​​WM\_LBUTTONDOWN​​：鼠标左键按下时触发。

显示弹窗，并在窗口上绘制文本（直接使用 GetDC + ReleaseDC）。

​​WM\_CLOSE：用户点击关闭按钮时触发。

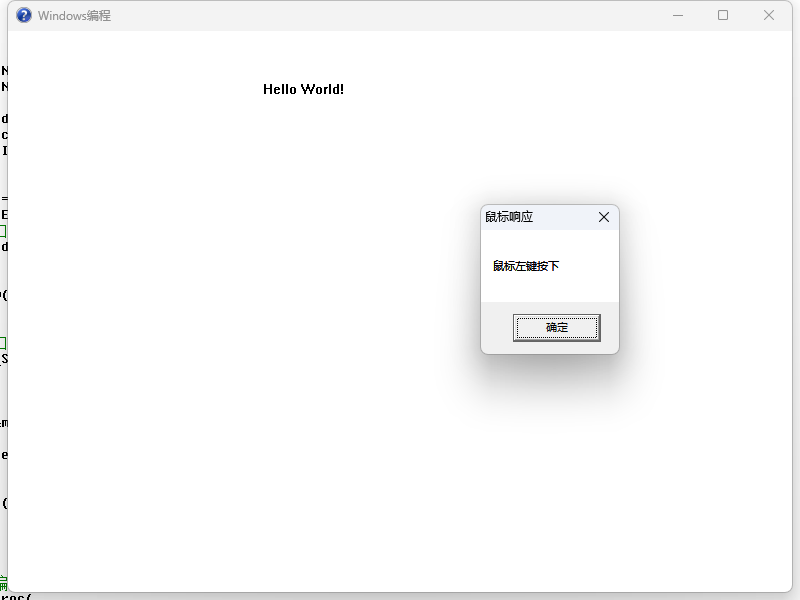
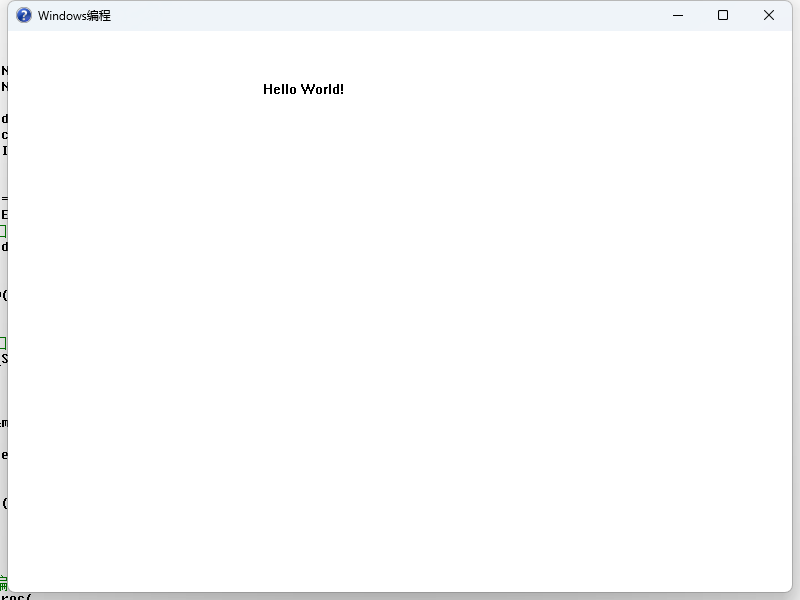
弹出确认对话框，确认后销毁窗口（DestroyWindow）。

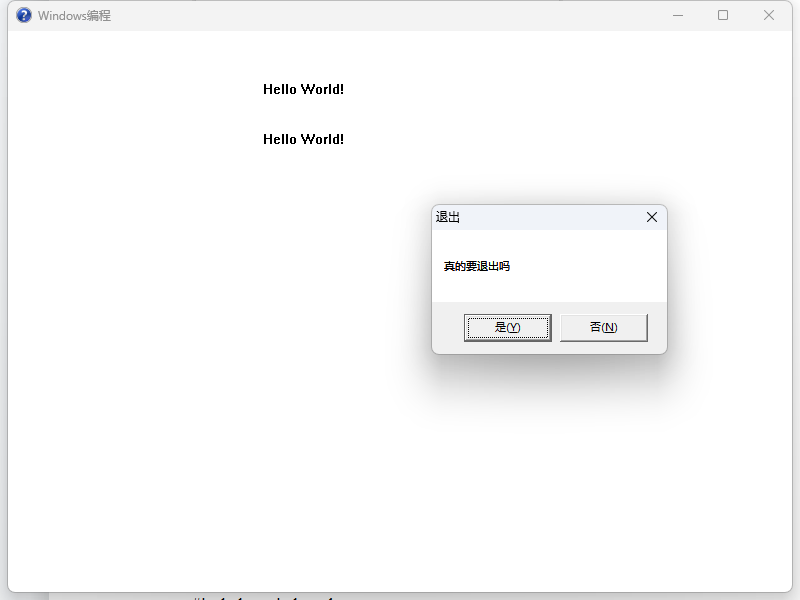
​​WM\_DESTROY​​：窗口被销毁时触发。

调用 PostQuitMessage 退出消息循环。

​​默认处理：未处理的消息交给 DefWindowProc（默认窗口过程）。

1. **实验结果与分析**





程序运行之后出现一个窗口，点击鼠标后会检测到鼠标左键按下，点击退出之后，会弹出是否确认推出

1. **源程序**

#include <windows.h>

#include <stdio.h>

LRESULT CALLBACK MyWndProc(

HWND hwnd,

UINT uMsg,

WPARAM wParam,

LPARAM lParam

);

//1.入口函数

int WINAPI WinMain(

HINSTANCE hInstance,

HINSTANCE hPrevInstance,

LPSTR lpCmdLine,

int nCmdShow

)

{

//2 创建窗口

//2.1设计一个窗口类

WNDCLASS MyWnd;

MyWnd.cbClsExtra = NULL;

MyWnd.cbWndExtra = NULL;

MyWnd.hbrBackground = (HBRUSH)GetStockObject(WHITE\_BRUSH);

MyWnd.hCursor = LoadCursor(NULL, IDC\_CROSS);

MyWnd.hIcon = LoadIcon(NULL,IDI\_QUESTION);

MyWnd.hInstance = hInstance;

MyWnd.lpfnWndProc = MyWndProc;

MyWnd.lpszClassName = "Hello";

MyWnd.lpszMenuName = NULL;

MyWnd.style = CS\_HREDRAW | CS\_VREDRAW;

//2.2 对设计好的窗口类进行注册

RegisterClass(&MyWnd);

//2.3 创建窗口

HWND hWnd;

hWnd = CreateWindow("Hello","Windows编程",WS\_OVERLAPPEDWINDOW,

0,0,800,600,

NULL,NULL,hInstance,NULL);

//2.4 显示及更新窗口

ShowWindow(hWnd,SW\_SHOW);

UpdateWindow(hWnd);

//3 消息循环

MSG msg;

while (GetMessage(&msg,NULL,0,0)) //从消息队列中获取消息

{

TranslateMessage(&msg); //消息解释

//发送消息

DispatchMessage(&msg); //将消息发送到“窗口过程”

}

return 0;

}

//4.“窗口过程函数”（编写消息响应代码）

LRESULT CALLBACK MyWndProc(

HWND hwnd,

UINT uMsg,

WPARAM wParam,

LPARAM lParam

)

{

switch (uMsg)

{

case WM\_PAINT:

PAINTSTRUCT ps;

{HDC hpaintDC = BeginPaint(hwnd,&ps);

TextOut(hpaintDC,255,50,"Hello World!",strlen("Hello World!"));

EndPaint(hwnd,&ps);}

break;

case WM\_CHAR:

char str[255];

sprintf(str,"char is %d",wParam);

MessageBox(hwnd,str,"按键响应",0);

break;

case WM\_LBUTTONDOWN:

MessageBox(hwnd,"鼠标左键按下","鼠标响应",0);

HDC hDC;

hDC = GetDC(hwnd);

TextOut(hDC,255,100,"Hello World!",strlen("Hello World!"));

ReleaseDC(hwnd,hDC);

break;

case WM\_CLOSE:

if (IDYES == MessageBox(hwnd,"真的要退出吗","退出",MB\_YESNO))

DestroyWindow(hwnd);

break;

case WM\_DESTROY:

PostQuitMessage(0);

break;

default:

return DefWindowProc(hwnd,uMsg,wParam,lParam);

}

return 0;

}