Como criar uma aplicação Windows usando WinMain

WinMain é a função principal de entrada para uma aplicação Windows escrita em C ou C++. Esta função é crucial para o desenvolvimento de aplicações que utilizam a API do Windows, pois ela inicializa a aplicação, cria a janela principal, e entra no loop de mensagens que permite à aplicação responder a eventos do sistema. Entender como usar WinMain é fundamental para qualquer desenvolvedor que deseja criar aplicações nativas para o ambiente Windows.

Exemplo 1: Estrutura Básica de um Programa com WinMain

Aqui está um exemplo básico de como uma aplicação Windows pode ser estruturada usando WinMain:

```
#include <windows.h>

// Prototipo da função WindowProc
LRESULT CALLBACK WindowProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM lP
aram);

int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpC
mdLine, int nCmdShow) {
    // Registro da classe da janela
    const char CLASS_NAME[] = "Sample Window Class";

    WNDCLASS wc = {};
    wc.lpfnWndProc = WindowProc;
    wc.hInstance = hInstance;
    wc.lpszClassName = CLASS_NAME;

    RegisterClass(&wc);

    // Criação da janela
    HWND hwnd = CreateWindowEx(
        0,                              // Estilo opcional da janela
        CLASS_NAME,                     // Nome da classe da janela
        "Sample Window",                // Texto da barra de título da jan
ela
        WS_OVERLAPPEDWINDOW,            // Estilo da janela

        // Tamanho e posição da janela
        CW_USEDEFAULT, CW_USEDEFAULT, CW_USEDEFAULT, CW_USEDEFAULT,

        NULL,       // Janela pai
        NULL,       // Menu
```

```
        hInstance,  // Instância do programa
        NULL        // Parâmetros de criação adicionais
    );

    if (hwnd == NULL) {
        return 0;
    }

    ShowWindow(hwnd, nCmdShow);

    // Loop de mensagens
    MSG msg = {};
    while (GetMessage(&msg, NULL, 0, 0)) {
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }

    return 0;
}

// Função de processamento de mensagens da janela
LRESULT CALLBACK WindowProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM lP
aram) {
    switch (uMsg) {
        case WM_DESTROY:
            PostQuitMessage(0);
            return 0;

        case WM_PAINT:
            PAINTSTRUCT ps;
            HDC hdc = BeginPaint(hwnd, &ps);
            FillRect(hdc, &ps.rcPaint, (HBRUSH) (COLOR_WINDOW+1));
            EndPaint(hwnd, &ps);
            return 0;
    }
    return DefWindowProc(hwnd, uMsg, wParam, lParam);
}
```

Exemplo 2: Compilando e Executando via CMD

Para compilar e executar este programa, você precisará do compilador cl.exe do Visual Studio. Siga os passos abaixo:

1. Abra o "Developer Command Prompt for Visual Studio".
2. Navegue até o diretório onde o arquivo de código-fonte está salvo.
3. Compile o código usando o comando:

```
cl /EHsc /W4 /DUNICODE /D_UNICODE your_program.c user32.lib gdi32.lib
```

4. Execute o programa resultante:

```
your_program.exe
```