

Software para Windows

Módulo 9 Aula 5

Linguagem C, o Curso Definitivo WR Kits

Autor: Dr. Eng. Wagner Rambo

Breve introdução

Para desenvolver um software para Windows, trabalharemos com níveis mais altos de abstração, pois temos a facilidade da biblioteca *windows.h*, disponível na maioria dos compiladores para esta plataforma. Programas Windows têm interfaces mais amigáveis e são em geral fáceis para o usuário final compreender, mesmo para quem nunca executou um programa na vida. O propósito da organização em janelas é oferecer um conceito que todos têm familiaridade, que é o de uma mesa de trabalho, onde você organiza seus arquivos em pastas, blocos de anotações, uma calculadora, etc. Você facilmente move os objetos de um lugar para o outro, ou eventualmente os joga na lixeira.

Softwares Windows trabalham muito bem com o mouse do computador, onde uma mensagem de recepção do clique do mouse será enviada para o seu programa, mas também permitem a utilização de teclado; sendo também baseados em ícones, com uma imagem simples que torna mais fácil de assimilarmos a aplicação de determinado programa. Menus, barra de ferramenta, barra de status e caixa de diálogo são mais recursos disponíveis e fáceis para o usuário final interagir.

O ambiente Windows é acessado por APIs (*Application Program Interface*), ou interface de programas aplicativos, que são basicamente um conjunto de centenas de funções que podem ser chamadas pelo seu programa.

Conceitos comuns aos softwares Windows

A função que inicia a execução de softwares Windows é a *WinMain()*, que é compilada utilizando a convenção WINAPI. A função *WinMain* retorna um inteiro (*int*). Para que o Windows se comunique com o seu programa, existe uma função de janela, que receberá uma mensagem em seus parâmetros. Todas as funções janela retornam o tipo *LRESULT CALLBACK*. *LRESULT* é um *typedef*. Qualquer função que é chamada pelo Windows é referenciada como uma função de *callback*. Normalmente um comando *switch* é utilizado para verificar respostas específicas para cada mensagem do qual o programa apresenta uma resposta.

Tipos de Dados Windows

Todos os tipos de dados usados pelo Windows foram convertidos em *typedef* na biblioteca *windows.h* e arquivos relacionados. Confira os principais tipos na Tabela 1.

Tipo	Descrição
HANDLE	Inteiro de 32 bits. Valor que identifica um recurso. Todo o tipo <i>handle</i> começa com H.
HWND	Inteiro de 32 bits. Usado como <i>handle</i> de janela.
BYTE	Inteiro sem sinal de 8 bits.
WORD	Inteiro sem sinal de 16 bits.
DWORD	Inteiro sem sinal longo.
UINT	Inteiro sem sinal de 32 bits.
LONG	É o mesmo que <i>long</i> dos tipos de dados do C.
BOOL	Inteiro que armazena 1 bit. <i>TRUE</i> , <i>FALSE</i> .
LPSRT	Ponteiro para uma <i>string</i> .
LPCSTR	Ponteiro constante para uma <i>string</i> .
Estrutura	Descrição
MSG	Mensagem do Windows.
WNDCLASS	Define uma classe de janela.

Tabela 1 - Tipos de dados básicos do Windows.

O programa winsoft, que está disponível no pacote de arquivos desta aula, apresenta o código mínimo para a criação de um programa em janela padrão Windows. A seguir algumas opções para modificar esta padronização em seus futuros softwares, que é realizada na definição e registro de classe de janela. Na Tabela 2 apresentamos as macros de ícones disponíveis.

Macro de ícone	Forma
IDI_APPLICATION	Ícone padrão (linha 50 e 51 do código).
IDI_ASTERISK	Ícone de informação.
IDI_EXCLAMATION	Ícone de ponto de exclamação.
IDI_HAND	Ícone de "Pare".
IDI_QUESTION	Ícone de ponto de interrogação.

Tabela 2 - Macros de ícones para a janela do software Windows.

Também é possível alterar o aspecto do cursor do mouse em seu programa, para isso, algumas das macros disponíveis estão explícitas na Tabela 3.

Macro de cursor	Forma
IDC_ARROW	Ponteiro de seta padrão (linha 52 do código).
IDC_CROSS	Cruz.
IDC_IBEAM	I vertical.
IDC_WAIT	Carregando.

Tabela 3 - Algumas macros de cursor para o mouse em seu programa Windows.

Após definir e registrar sua classe de janela, a aplicação irá criar uma janela com a função *CreateWindow()*, que tem o seu protótipo exibido no Box 1.

```
HWND CreateWindow(  
    LPCSTR    lpClassName,    /* nome da classe da janela */  
    LPCSTR    lpwinName,     /* título da janela */  
    DWORD     dwStyle,        /* tipo da janela */  
    int       X,              /* coordenada X do canto sup. esq. */  
    int       Y,              /* coordenada Y do canto sup. esq. */  
    int       width,          /* largura da janela */  
    int       height,         /* altura da janela */  
    HWND      hParent,        /* handle da janela-mãe */  
    HMENU     hMenu,          /* handle do menu principal */  
    HINSTANCE hThisInst,      /* handle da instância criadora */  
    LPVOID     lpzAdditional /* ponteiro para informação adicional */  
);
```

Box 1 - Protótipo da função *CreateWindow*.

Após criar a janela, é chamada a função *ShowWindow* para de fato exibir a janela. O código entre em loop para exibição das mensagens com *GetMessage*, onde *TranslateMessage* traduz os códigos de teclas virtuais utilizadas para o Windows, permitindo a interação do teclado com seu programa. A mensagem é enviada de volta para o Windows através da função *DispatchMessage*, que manterá a mensagem até que ela possa ser de fato passada para a função de janela do programa. A função de janela *WindowProcedure* responde apenas a uma única mensagem visível: *WM_DESTROY*, mensagem enviada quando o programa é encerrado. Quaisquer outras mensagens recebem o tratamento padrão através de *DefWindowProc*.

Bibliografia:

DAMAS, Luís; Linguagem C, décima edição.

Disponível em: <https://amzn.to/3nGdlbN>

SCHILDT, Herbert; C Completo e Total, terceira edição.

Disponível em: <https://amzn.to/3xxi3l8>