
Sistemas Operativos 2

2021/22

Aplicações nativas Win32/WinNT com UI gráfico

DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

1

Tópicos

Aplicações gráficas Win32
Modelo de programação por eventos
Tratamento de mensagens e eventos
Recursos no modelo de programação Windows

Bibliografia específica para este capítulo:

- Advanced Windows (3rd Edition); Jeffrey Richter
- WindowsNT 4 Programming; Herbert Schildt
- MSDN Library – PlatformSDK: DLLs, Processes, and Threads (disponível *online* e no ISEC)

DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

2

Windows NT – Modelo(s) de programação

Aplicações Win32

- Constituem o modelo principal de aplicações nativas para windows
- Modelo computacional: “Win API” (Win32 ou Win64)
- Win32 e Win64: Essencialmente o mesmo conjunto de funções
 - Apenas tamanho de dados e de ponteiros varia
 - Código de 32 bits é compatível com aplicações 64 bits
 - “Win32 API” → “Win API”

Alternativas ao Win API:

- Frameworks tais como MFC ou .NET
- Oferecem um conjunto de funcionalidade já implementada
- Não cobrem toda a funcionalidade do windows, pelo que é imprescindível entender a lógica Win API

DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

3

Windows NT – Modelo(s) de programação

O API Win32 é a base para diversos modelos de programação

- Um ou mais por cada um dos sub-sistemas

Exemplos

- *Win32 console application* – Aplicações CUI (*Console User Interface*)
- *Win32 application* – Aplicações GUI (*Graphical User Interface*)
- **Ambos directamente suportados pelo subsistema win32**

Win32 console application ← Já abordadas

- Programação sequencial seguindo os mesmos paradigmas que os programas consola
- Interacção com o utilizador via consola

Win32 Application ← Foco das próximas aulas

- Programação orientada por eventos, em que a sequência do programa deixa de ser directamente controlada pelo programador
- Interacção como utilizador via interface gráfica

DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

4

Windows NT – Win32 application

Alguns conceitos chave nesta parte da matéria

- Eventos
- Programação orientada a eventos
- Ciclo de processamento de mensagens / eventos
- Janelas
- Função *callback* de atendimento de mensagens
- Recursos de programas

DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

5

Windows NT – Win32 application

Função principal do programa

```
int WINAPI WinMain (  
    HINSTANCE hInstance,      // handle da instância actual  
    HINSTANCE hPrevInstance,  // handle da inst. anterior  
    LPSTR lpCmdLine,          // ptr para a linha de comando  
    int nCmdShow               // estado inicial da janela  
);
```

(hPrevInstance é sempre NULL na família NT)

DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

6

Windows NT – Win32 application

Char vs. WCHAR → Abordagem habitual → `_t` / TCHAR

```
_tWinMain

#ifdef _UNICODE
#define _tWinMain wWinMain
#else
#define _tWinMain WinMain
#endif
```

Ou seja:
-> Usar `_tWinMain`

DEIS/SEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

7

Windows NT – Win32 application

Não é obrigatório haver janelas num programa Win32

Exemplo: programa mais pequeno que se pode fazer em Win32: não faz nada e não cria janela nenhuma

```
#include <windows.h>

int WINAPI WinMain (
    HINSTANCE hInstance, HINSTANCE hPrevInstance,
    LPSTR LpCmdLine, int nCmdShow) {
    /* não faz nada */
    return 1;
};
```

DEIS/SEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

8

Windows NT – Win32 application

Exemplo: programa que mostra os parâmetros da linha de comandos

É necessária uma janela para apresentar os dados

Alternativas possíveis:

- Janela definida pelo programador
- Janela segundo um modelo já existente no Windows

O Windows/API Win32 oferece uma grande variedade de recursos para interface gráfico:

- Janelas para fins específicos (Exemplo: dialog boxes)
- Controlos (menus, botões, check-boxes, etc.)

→ Para este exemplo: usa-se um modelo já existente: MessageBox

DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

9

Windows NT – Win32 application

Função para apresentar uma *Message Box*

```
int WINAPI MessageBoxEx(  
    HWND hWnd,          // Janela pai/dono (opcional)  
    LPCTSTR lpText,      // texto / informação (opcional)  
    LPCTSTR lpCaption,   // Título (opcional)  
    UINT uType,          // Tipo (implica botões específicos)  
    WORD wLanguageId    // Código língua (0 = de sistema)  
);
```

Tipo:

MB_OK, MB_OKCANCEL, MB_YESNO, etc.

Retorna (consoante o tipo especificado e botões accionados):

IDOK, IDYES, IDCANCEL, IDABORT, IDNO, IDRETRY, etc.

DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

10

Windows NT – Win32 application

Programa “Hello World” que mostra uma *Message Box* com os parâmetros da linha de comandos

```
#include <windows.h>
int WINAPI WinMain (
    HINSTANCE hInstance, HINSTANCE hPrevInstance,
    LPSTR lpCmdLine, int nCmdShow) {
    MessageBoxEx(NULL, lpCmdLine, "Hello World",
        MB_OK | MB_ICONINFORMATION | MB_TASKMODAL, 0);
    return 0;
}
```

Funções com sufixo “Ex” no nome (não existe *overloading* no Win32):

→ São versões *melhoradas* de funções já existentes

DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

11

Windows NT – Win32 application

Exemplo2: programa que mostra uma *MessageBox* com o *pathname* usado na execução do programa

```
#include <windows.h>
int WINAPI WinMain (
    HINSTANCE hInstance, HINSTANCE hPrevInstance,
    LPSTR lpCmdLine, // ver -> GetCommandLine
    int nCmdShow) {
    // int argc */
    // char ** __argv -> __wargv ou __targv + TCHAR.h
    MessageBoxEx(NULL, __argv[0], "Command Line[0]",
        // -> L"... " -> _T("...") -> _T("Command Line[0]")
        MB_OK | MB_ICONINFORMATION | MB_TASKMODAL,
        0);
    return 1;
};
```

DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

12

Windows NT – Win32 application

Normalmente um programa Win32 utiliza janelas e elementos gráficos (com a possível excepção de aplicações de consola).

Estes programas seguem a lógica da **programação por eventos**

Esquema de um programa Win32 que usa janelas:

Na função WinMain

1. Definir classe (tipo) de janela
2. Registar classe de janela (associada a uma função de janela)
3. Criar a(s) janela(s)
4. Mostrar a(s) janela(s)
5. Ciclo de tratamento de mensagens

Na função “da janela”:

Tratamento dos eventos enviados à janela

Eventos: sempre tratados no contexto da thread que criou a janela

DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

13

Windows NT – Win32 application

Definição e registo de uma classe de janelas

```
typedef struct _WNDCLASSEX {  
    UINT cbSize;           // tamanho da estrutura  
    UINT style;            // estilo da janela  
    WNDPROC lpfnWndProc;   // função da janela  
    int cbClsExtra;        // bytes extra para info da classe  
    int cbWndExtra;        // bytes extra para info da janela  
    HANDLE hInstance;      // instância do programa  
    HICON hIcon;           // icon std. associado à janela  
    HICON hIconSm;         // icon peq. associado à janela  
    HCURSOR hCursor;       // cursor associado à janela  
    HBRUSH hbrBackground;  // "côr" de fundo da janela  
    LPCTSTR lpszMenuName;  // nome do rec menu (se houver)  
    LPCTSTR lpszClassName; // nome para esta classe  
} WNDCLASSEX;             // esta struct JÁ EXISTE – só USAR
```

Para registar a classe utiliza-se a função **RegisterClassEx**

→ **Atenção à questão dos char/wchar_t**

DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

14

Windows NT – Win32 application

Criação de uma janela

```
HWND CreateWindow(  
    LPCTSTR lpClassName,    // pont. para o nome da classe  
    LPCTSTR lpWindowName,    // título da janela  
    DWORD dwStyle,          // estilo da janela  
    int x,                   // posição horizontal da janela  
    int y,                   // posição vertical da janela  
    int nWidth,              // largura da janela  
    int nHeight,             // altura da janela  
    HWND hWndParent,         // handle para a janela "pai"  
    HMENU hMenu,             // handle para o menu  
    HANDLE hInstance,        // handle da instancia do prog.  
    LPVOID lpParam           // ponteiro para dados adicionais  
);
```

Nota: LPCSTR ou LPCWSTR, dependendo se o projecto usa char ou wchar_t
→ É melhor usar sempre TCHAR / TCHAR *, TEXT("...") em vez de char/wchar_t

DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

15

Windows NT – Win32 application

Protótipo e estrutura da função da janela

```
LRESULT CALLBACK WindowFunc(  
    HWND hwnd,               // handle da janela  
    UINT msg,                 // código da mensagem  
    WPARAM wParam,           // informação relativa à mensagem  
    LPARAM lParam            // mais informação relativa à msg.  
) {  
    switch(msg) {  
        case WM_****:  
            ...  
            break  
        default:  
            return DefWindowProc(hwnd, msg, wParam, lParam);  
    }  
    return 0;  
}
```

DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

16

Windows NT – Win32 application

Ciclo de tratamento das mensagens

```
while(GetMessage(&msg, NULL, 0, 0)) {    /* OS -> app */  
    TranslateMessage(&msg);  
    DispatchMessage(&msg);              /* -> janela */  
}  
  
return msg.wParam;
```

- Quando não existem mensagens para serem processadas o programa não ocupa tempo do processador
- O processamento das mensagens pela função da janela deve ser o mais breve possível. Se for necessário efectuar um processamento demorado este deve ser efectuado por uma *thread* lançada para esse efeito (ficando a *thread* original livre para processar outras mensagens)

DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

17

Windows NT – Win32 application

Programa exemplo – 1) Início da função principal do programa

```
int WINAPI WinMain(HINSTANCE hThisInst, HINSTANCE hPrevInst,  
    LPSTR lpszCmdLine, int nWinMode) {  
  
    TCHAR JanelaPrinc[] = TEXT("Rects"); // nome classe da janela  
    HWND hwnd;  
    MSG msg;  
  
    if (!RegistaClasse(hThisInst, JanelaPrinc)) // (mais adiante)  
        return 0;  
    hwnd = CriarJanela(hThisInst, JanelaPrinc); // (mais adiante)  
  
    /* Apresenta a janela no ecran */  
  
    ShowWindow(hwnd, nWinMode);  
    UpdateWindow(hwnd);
```

DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

18

Windows NT – Win32 application

Programa exemplo – 2) Resto da função principal do programa

```
/* Ciclo de mensagens */

while(GetMessage(&msg, NULL, 0, 0)) {
    TranslateMessage(&msg);    /* traduz código teclado */
    DispatchMessage(&msg);    /* reencaminha para a janela alvo */
}

return msg.wParam;
}
```

```
// GetMessage:
// NULL -> p/ qualquer janela
// Filtro: 1ª msg - ex. WM_KEYFIRST
// Filtro: última msg - ex. WM_KEYLAST
```

O ciclo termina quando é recebida uma mensagem WM_QUIT

DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

19

Windows NT – Win32 application

Função RegistaClasse

```
ATOM RegistaClasse(HINSTANCE hThisInst, TCHAR * szWinName) {
    WNDCLASSEX wcl;

    wcl.cbSize = sizeof(WNDCLASSEX);
    wcl.hInstance = hThisInst;    // handle para esta instância
    wcl.lpszClassName = szWinName;    // nome da classe
    wcl.lpfnWndProc = WindowFunc;    // função da janela
    wcl.style = CS_HREDRAW;    // estilo default+
    // CS_GLOBALCLASS -> classe global em vez de local
    wcl.hIcon = LoadIcon(NULL, IDI_APPLICATION);    // icon std
    wcl.hIconSm = LoadIcon(NULL, IDI_WINLOGO);    // icon menor
    // wcl.hIconSm = LoadIcon(hThisInst, (LPCTSTR)IDI_ICON1);
    wcl.hCursor = LoadCursor(NULL, IDC_ARROW);    // cur. style
    wcl.lpszMenuName = NULL;    // sem menu
    wcl.cbClsExtra = 0;    // sem info extra
    wcl.cbWndExtra = 0;    // sem info extra
    wcl.hbrBackground = (HBRUSH) GetStockObject(WHITE_BRUSH);
    // cor de fundo = branco

    return RegisterClassEx(&wcl);
}
```

DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

20

Windows NT – Win32 application

Função CriarJanela

```
HWND CriarJanela(HINSTANCE hThisInst, TCHAR * szWinName) {  
  
    return CreateWindow(  
        szWinName,                // nome da classe desta janela  
        TEXT("S02 - exemplo Win32"), // título  
        WS_OVERLAPPEDWINDOW,      // estilo da janela = normal  
        CW_USEDEFAULT,            // coordenada x = escolhida pelo windows  
        CW_USEDEFAULT,            // coordenada y = escolhida pelo windows  
        CW_USEDEFAULT,            // largura = escolhida pelo windows  
        CW_USEDEFAULT,            // altura = escolhida pelo windows  
        HWND_DESKTOP,             // sem janela pai  
        NULL,                     // sem menu  
        hThisInst,                // handle para esta instancia do programa  
        NULL,                     // sem argumentos (informação) adicional  
    );  
}
```

DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

21

Windows NT – Win32 application

Programa exemplo – Dados necessários ao programa

Exemplo concreto: este programa vai desenhar rectângulos no ecrã

```
/* estrutura de dados para representar um rectângulo */  
typedef struct {  
    int xi, yi, xf, yf;  
} Rect;  
  
#define MAX_RECTS 250  
  
Rect mat[MAX_RECTS];  
int numrect;        /* número actual de rectângulo */  
int drawing;        /* estado a desenhar / não está a desenhar */  
  
int x1, y1, x2, y2; /* coordenadas do rectângulo actual */  
  
// POINT p;
```

DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

22

Windows NT – Win32 application

Função *callback* da janela (1)

```
LRESULT CALLBACK WindowFunc(HWND hwnd, UINT message,
                             WPARAM wParam, LPARAM lParam) {
    static TCHAR * msg = TEXT("Desenho de Rectângulos");
    PAINTSTRUCT ps;
    HDC hdc;
    int i;

    switch(message) {
    case WM_CREATE:
        drawing = 0;
        numrect = 0;
        break;
    case WM_DESTROY:      /* Termina o programa */
        PostQuitMessage(0); // 0 é o exit-code
        break;
    }
```

DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

23

Windows NT – Win32 application

Função *callback* da janela (2)

```
case WM_LBUTTONDOWN:
    x1 = LOWORD(lParam);      /* posição horizontal */
    y1 = HIWORD(lParam);      /* posição vertical */
    drawing = 1;              /* assinala linha em construção */

case WM_MOUSEMOVE:
    if (drawing==1) {
        x2 = LOWORD(lParam);      /* posição horizontal */
        y2 = HIWORD(lParam);      /* posição vertical */
        InvalidateRect(hwnd, NULL, true); /* requisita WM_PAINT */
    }
    break;
```

DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

24

Windows NT – Win32 application

Função *callback* da janela (3)

```
case WM_LBUTTONDOWN:
    if (drawing==1) {
        x2 = LOWORD(lParam);           /* posição horizontal */
        y2 = HIWORD(lParam);           /* posição vertical */
        mat[numrect].xi = x1;
        mat[numrect].yi = y1;
        mat[numrect].xf = x2;
        mat[numrect].yf = y2;
        if (numrect < MAX_RECTS - 1)
            numrect++;
        drawing = 0;
        InvalidateRect(hwnd,NULL,true);
    }
    break;
```

DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

25

Windows NT – Win32 application

Função *callback* da janela (4)

```
case WM_PAINT:
    hdc = BeginPaint(hwnd, & ps);
    TextOut(hdc, 50, 50, msg, _tcslen(msg)); // strlen _tcslen
    for (i=0; i<numrect; i++) {
        Rectangle(hdc, mat[i].xi, mat[i].yi, mat[i].xf, mat[i].yf);
    }
    if (drawing==1) {
        Rectangle(hdc, x1, y1, x2, y2);
    }
    EndPaint(hwnd, &ps);
    break;
```

DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

26

Windows NT – Win32 application

Função *callback* da janela (5)

```
default:
    /* as mensagens que não são explicitamente tratadas
       pelo programa são deixadas ao tratamento default
       através da função DefWindowProc */
    return DefWindowProc(hwnd, message, wParam, lParam);
}
return 0;
} /* fim da função da janela */
```

- O tratamento das mensagens não deve ser demorado de forma a permitir à aplicação obter e tratar as mensagens seguintes
- Se for necessário um processamento demorado em resposta a uma mensagem, deve ser lançada uma *thread* para esse processamento

DEIS/ISEC

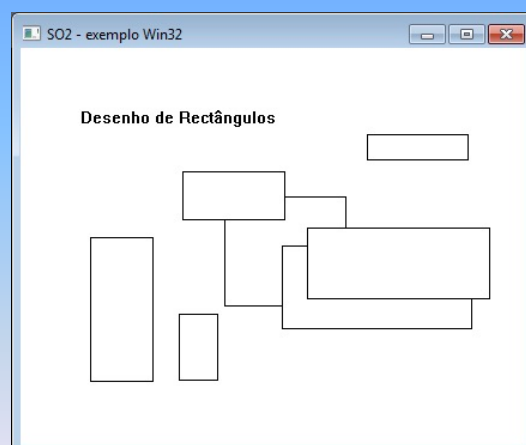
Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

27

Windows NT – Win32 application

Aspecto do programa-exemplo em execução:



DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

28

Windows NT – *Win32 application*

E se por acaso se quisesse ter duas janelas (ou mais)?

As janelas fazem o mesmo?

- Sim:
 - Mesma classe e mesma função de janela
 - Os dados são os mesmos?
 - Sim:
 - Não é preciso fazer mais nada
 - Não:
 - É preciso distinguir o contexto de execução da função da janela
- Não:
 - Nova classe de janela e com nova função de janela

DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

29

Windows NT – *Win32 application*

Duas janelas, mesmo trabalho e mesmos dados

Nova função main

```
// ...
HWND hwnd, hwnd2;

if (!RegistaClasse(hThisInst, JanelaPrinc))
    return 0;

hwnd = CriarJanela(hThisInst, JanelaPrinc);
hwnd2 = CriarJanela(hThisInst, JanelaPrinc);

/* Muda títulos (só para diferenciar melhor) */

SetWindowText(hwnd, TEXT("Janela 1"));
SetWindowText(hwnd2, TEXT("Janela 2"));
```

DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

30

Windows NT – Win32 application

Duas janelas, mesmo trabalho e mesmos dados

```
/* Apresenta as janelas no ecran */

ShowWindow(hwnd, nWinMode);
UpdateWindow(hwnd);

ShowWindow(hwnd2, nWinMode);
UpdateWindow(hwnd2);

/* Ciclo de mensagens -> é o mesmo - só um */

while(GetMessage(&msg, NULL, 0, 0)) {
    TranslateMessage(&msg); /* traduz código teclado */
    DispatchMessage(&msg); /* reencaminha para a janela alvo */
}
return msg.wParam;
}
```

DEIS/ISEC

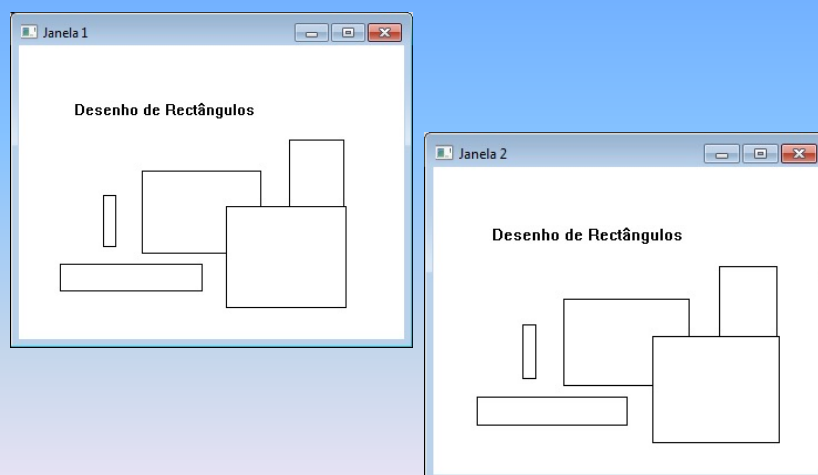
Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

31

Windows NT – Win32 application

Aspecto do programa-exemplo com duas janelas



DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

32

Windows NT – *Win32 application*

O processamento de eventos deve ser o mais breve possível.

- As janelas têm o seu atendimento de eventos executado no contexto da thread onde foram criadas (mesmo que sejam de classes diferentes).
- Se o atendimento de um evento for demorado (ou encalhar), o atendimento de todos os eventos de todas as janelas lançadas nessa thread ficam por executar.
- Se houver necessidade de processamento longo no atendimento de um evento, deve-se lançar uma thread nova para fazer esse processamento.

DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

33

Windows NT – *Win32 application*

Experiência: acrescentar o seguinte tratamento de eventos do teclado

→ Sempre que se carrega na tecla 'e' ('e' de "en^{cal}har") entra-se num ciclo infinito (ter o task maneger a postos)

Na função da janela:

```
case WM_CHAR:
    if (wParam == 'e')
        while(1);    // ciclo infinito
    break;
...
```

- Independentemente da janela em que se carregou em "e", ambas as janelas ficam com uma aparência congelada: a *thread* responsável por tratar os seus eventos é a mesma e está presa no ciclo infinito
- Se as janelas pertencessem a threads diferentes, uma janela ficaria funcional, mas aquela em que se carregou em 'e' continuaria congelada.

DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

34

Windows NT – *Win32 application*

No contexto do exemplo das duas janelas

- Ambas as janelas apresentam os mesmos dados e comportamento
- Como conseguir diferentes dados em diferentes janelas da mesma classe ?

→ Passar as variáveis que armazenam os dados para a função da janela?

- Não iria funcionar pois o valor dessas variáveis perde-se quando a função da janela termina
- Não esquecer: a função da janela é invocada (executada e terminada) de cada vez que ocorre um evento associado à janela

DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

35

Windows NT – *Win32 application*

Resolução (existem outras soluções, algumas mais complexas)

- Os dados devem existir fora da função da janela.
- Para cada janela deve existir um conjunto completo dos dados em questão
- Na função da janela (por exemplo, em WM_PAINT) deve-se averiguar a que janela é que a função está a responder (handle) e, em função dessa informação, ir buscar o conjunto de dados dados adequados.
- Os dados devem ser acessíveis de dentro da função da janela

Pseudo-código na função da janela, exemplo para o caso de WM_PAINT

```
case WM_PAINT:    // ou outro evento
    sou a janela (handle) X, logo uso os dados associados a essa janela

    /* sei o handle, mas como chego daí aos dados da janela cujo handle
    é X? */
```

DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

36

Windows NT – *Win32 application*

Como chegar aos dados da janela em questão a partir do handle?

- Duas estratégias

1ª estratégia

- Colocar os dados em questão numa variável estruturada.
- Colocar na estrutura do Windows que descreve a janela um ponteiro para essa variável estruturada com os dados em questão (reservando espaço para tal na estrutura WNDCLASSEX)
 - Na criação da janela apenas é necessário indicar o ponteiro para a variável.
 - A variável estruturada com os dados apenas tem que existir durante a vida da janela, mas não tem que ser uma variável global
- Na função de tratamento de eventos, obtém-se o ponteiro e acede-se à variável estruturada com os dados necessários relativos à janela
 - A janela guardar numa variável global ou (de acesso global) uma "lista" de estruturas (não se trata de uma "lista ligada").

Não estão envolvidas nem variáveis globais nem pesquisas demoradas

DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

37

Windows NT – *Win32 application*

1ª estratégia – API envolvido (3 passos)

1) Registrar o ponteiro para a estrutura com os dados nos bytes extra associados à janela

- Esse ponteiro pode ser obtido a partir da função da janela
- O ponteiro indica directamente onde está a estrutura com os dados. Não é preciso pesquisa nenhuma e a estrutura não precisa de ser global – basta manter-se em memória durante a vida da janela.

Definição na classe da janela (estrutura WNDCLASSEX)

```
wcl.cbWndExtra = sizeof(EstruturaDados *);
```

2) Colocação de um ponteiro (criação da janela / na função main)

```
EstruturaDados dados; ...  
SetWindowLongPtr(hwnd, 0, (LONG_PTR) &dados);
```

3) Obtenção do ponteiro (na função da janela)

```
EstruturaDados * pont = (EstruturaDados *)  
GetWindowLongPtr(hwnd, 0);
```

DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

38

Windows NT – *Win32 application*

Como obter os dados que dizem respeito à janela em questão?

2ª estratégia

- Guardar numa variável global ou (de acesso global) uma “lista” de estruturas (não se trata de uma “lista ligada”).
 - Cada elemento da lista é uma cópia dos dados de uma janela em particular
 - O handle da janela faz parte dos elementos guardados nessa lista
 - A função da janela percorre essa lista e compara o handle que lhe foi dado com os handles da lista até encontrar o elemento que lhe diz respeito

Desvantagens

- Envolve pesquisa (performance)
- Uso variáveis globais (péssimo)
- Mais código (preencher a lista, percorrer a lista etc.)

Esta segunda estratégia é usada num exemplo em slides seguintes apenas porque tem mais código que a primeira alternativa.

A primeira estratégia é a melhor e é a recomendada

DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

39

Windows NT – *Win32 application*

Dados necessários ao trabalho de cada thread:

- ID da janela (o handle), matriz de rectângulos, coordenadas de trabalho, estado

```
typedef struct {
    int xi,yi,xf,yf;
} Rect;

#define MAX_RECTS 250

struct InfoTrab {
    HWND janelaID;           // dados para uma janela
                             // handle da janela a que diz respeito
    Rect m1[MAX_RECTS];
    int numrect;
    int drawing;
    int x1,y1,x2,y2;
};

#define NUM_JANELAS 2

struct InfoTrab dados[NUM_JANELAS]; // global, acessível à função
```

DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

40

Windows NT – *Win32 application*

Função auxiliar: permite obter acesso aos dados relativos a uma janela

```
// devolve ponteiro para estrutura dados conforme handle fornecido

struct InfoTrab * getOsMeusDados(HWND id) {
    int i;
    for (i=0; i<NUM_JANELAS; i++)
        if (dados[i].janelaID == id)    // matriz já preenchida
            return & dados[i];
    // fim do ciclo => não encontrou. Vai retornar NULL
    return NULL;
}

// Apenas para exemplo porque precisa de mais código.
// Não esquecer: 1a estratégia (a outra) é a melhor!
```

A função é invocada a partir da função de tratamentos de eventos da janela recebendo o handle da janela que estiver a ser tratada.

Nota: Não é a solução mais eficiente mas é simples e serve para este exemplo

DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

41

Windows NT – *Win32 application*

Na função main é preciso preparar a matriz de estruturas com os dados de trabalho

```
dados[0].janelaID = hwnd;
dados[1].janelaID = hwnd2;
```

Os restantes dados são inicializados na WM_CREATE

DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

42

Windows NT – Win32 application

A função da janela é modificada da seguinte maneira:

→ Em alguns dos eventos é necessário ir buscar os dados relativos à janela no contexto da qual a função está a ser executada

Código afectado neste exemplo: tratamento das mensagens relacionadas com a manipulação e visualização dos dados guardados na estrutura InfoTrab:

- WM_CREATE,
- WM_LBUTTONDOWN
- WM_MOUSEMOVE
- WM_LBUTTONUP
- WM_PAINT

No tratamento destas mensagens irá existir um ponteiro **mydata** que apontará para o elemento da matriz dados que diz respeito à janela em questão. Esse ponteiro é obtido através da função **getOsMeusDados**

DEIS/SEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

43

Windows NT – Win32 application

Alterações à função da janela (WM_CREATE e WM_LBUTTONDOWN)

```
// ...
struct InfoTrab * mydata;

switch(message) {
    case WM_CREATE:
        mydata = getOsMeusDados(hwnd) ;
        mydata->drawing = 0;
        mydata->numrect = 0;
        break;

    case WM_LBUTTONDOWN:
        mydata = getOsMeusDados(hwnd) ;
        mydata->x1 = LOWORD(lParam) ;
        mydata->y1 = HIWORD(lParam) ;
        mydata->drawing = 1;
```

DEIS/SEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

44

Windows NT – Win32 application

Alterações à função da janela: WM_MOUSEMOVE

```
case WM_MOUSEMOVE:
    mydata = getOsMeusDados(hwnd);
    if (mydata->drawing==1) {
        mydata->x2 = LOWORD(lParam);
        mydata->y2 = HIWORD(lParam);
        InvalidateRect(hwnd,NULL,TRUE);
    }
    break;
```

DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

45

Windows NT – Win32 application

Alterações à função da janela: WM_LBUTTONDOWN

```
case WM_LBUTTONDOWN:
    mydata = getOsMeusDados(hwnd);
    if (mydata->drawing == 1) {
        mydata->x2 = LOWORD(lParam);
        mydata->y2 = HIWORD(lParam);
        mydata->mat[mydata->numrect].xi = mydata->x1;
        mydata->mat[mydata->numrect].yi = mydata->y1;
        mydata->mat[mydata->numrect].xf = mydata->x2;
        mydata->mat[mydata->numrect].yf = mydata->y2;
        if (mydata->numrect < MAX_RECTS - 1)
            mydata->numrect++;
        mydata->drawing = 0;
        InvalidateRect(hwnd,NULL,TRUE);
    }
    break;
```

DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

46

Windows NT – Win32 application

Alterações à função da janela: WM_PAINT

```
case WM_PAINT:
    mydata = getOsMeusDados(hwnd);
    hdc = BeginPaint(hwnd, &ps);
    TextOut(hdc, 50, 50, msg, _tcslen(msg));
    for (i=0; i<mydata->numrect; i++) {
        Rectangle(hdc, mydata->mat[i].xi, mydata->mat[i].yi,
            mydata->mat[i].xf, mydata->mat[i].yf);
    }
    if (mydata->drawing == 1) {
        Rectangle(hdc, mydata->x1, mydata->y1,
            mydata->x2, mydata->y2);
    }
    EndPaint(hwnd, &ps);
    break;
```

DEIS/ISEC

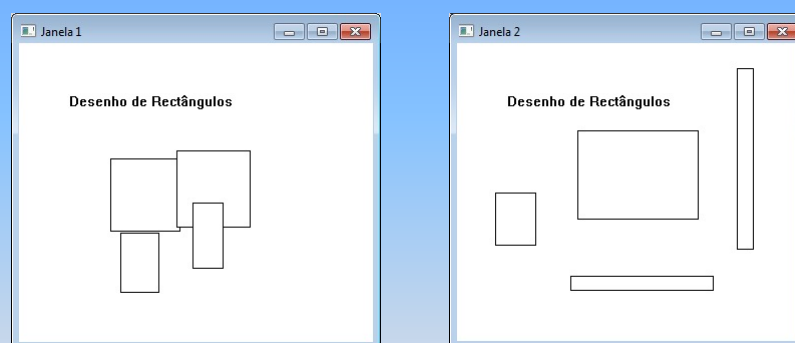
Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

47

Windows NT – Win32 application

Execução: agora cada janela trabalha com os seus dados



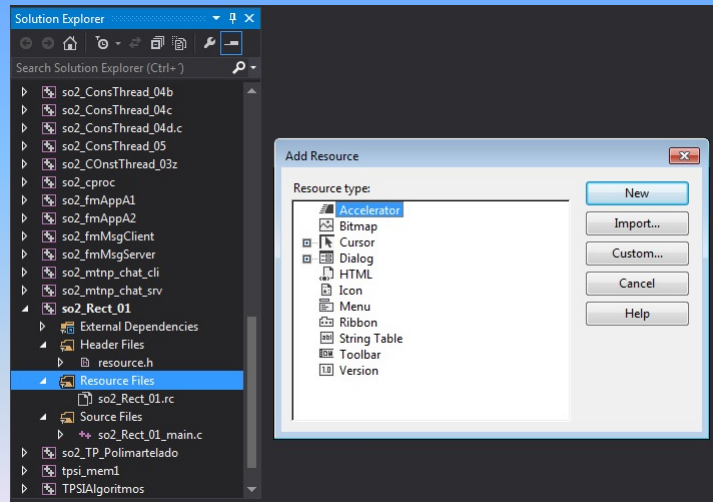
DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

48

Windows NT – Win32 resources



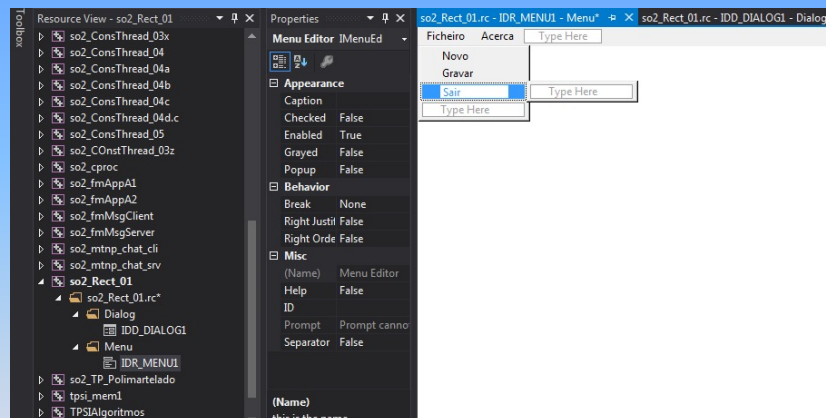
DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

49

Windows NT – Win32 application



DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

50

Windows NT – Win32 application

Resource.h

```
//{{NO_DEPENDENCIES}}
// Microsoft Developer Studio generated include file.
// Used by so2a.rc
//

#define IDI_ICON1                101

#define IDR_MENU1                102
#define ID_FICHEIRO_SAIR        40001
#define ID_FICHEIRO_ACERCA      40003

// ...
```

DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

51

Windows NT – Win32 application

- A construção dos recursos também pode ser feita manualmente

```
IDR_MENU1 MENU DISCARDABLE
BEGIN
    POPUP "&Ficheiro"
    BEGIN
        MENUITEM "&Acerca", ID_FICHEIRO_ACERCA
        MENUITEM "&Sair", ID_FICHEIRO_SAIR
    END
END
```

DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

52

Windows NT – Win32 application

Na definição da classe da janela (estrutura **WNDCLASSEX**):

```
wcl.lpszMenuName = NULL;
```



```
wcl.lpszMenuName = MAKEINTRESOURCE(IDR_MENU1);
```

Na função da janela é necessário apanhar a mensagem **WM_COMMAND**:

```
switch(msg) {
    // ... Outras mensagens
    case WM_COMMAND: // indica comando do menu
        switch(LOWORD(wParam)) { // qual foi o menu?
            case ID_FICHEIRO_ACERCA:
                // faz qualquer coisa + break /return
            case ID_FICHEIRO_EXIT:
                PostQuitMessage(0);
                return 0;
        }
    return 0;
}
```

DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

53

Windows NT – Win32 application

Criação da dialog box: Modal vs Modeless

- **Modal** – O controlo permanece na dialog-box

```
int DialogBox(
    hInstance, // instance handle
    MAKEINTRESOURCE(IDD_MAINDLG), // identificador rc
    hWndParent, // janela pai NULL=não mod.
    DlgProc); // função da janela da dlg

// se retornar -1 houve um erro
```

- **Modeless** – Continua a ser possível interagir com as outras janelas

```
HWND CreateDialog(
    hInstance, // instance handle
    MAKEINTRESOURCE(IDD_MAINDLG), // identificador rc
    hWndParent, // janela pai
    DlgProc); // função da janela da dlg

// retorna NULL se houve erro
ShowWindow(dlg, SW_SHOW);
// ...
IsDialogMessage(dlg, msg); // true se a msg foi processada
```

DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luís Nunes

54

Windows NT – Win32 application

Função *callback* da janela da dialog box

```
BOOL CALLBACK DlgProc(HWND dlg, UINT msg, WPARAM wParam,
                      LPARAM lParam) {

    char dlgStr[20];
    switch(msg) {
        case WM_INITDIALOG:
            return TRUE;
        case WM_COMMAND:
            switch(LOWORD(wParam)) {
                case IDOK:
                    if(GetDlgItemText(dlg, IDC_EDIT1, (LPTSTR)dlgStr, 20))
                        MessageBox(dlg, dlgStr, "Text Message", MB_OK);
                    return TRUE;
                default:
                    return TRUE;
            }
        case WM_CLOSE:
            EndDialog(dlg, IDOK); // janela criada com DialogBox()
            DestroyWindow(dlg);   // janela criado com CreateDialog()
            return TRUE;
    }
    return FALSE;
}
```

DEIS/ISEC

Sistemas Operativos 2 – 2021/22

João Durães, José Luis Nunes