

## Introducere in programarea sub sistemul de operare Windows

Deoarece sistemul de operare Windows este un system de operare multitasking programele vor partaja spatial de memorie al sistemului de calcul.

Din acest motiv trebuie avut in vedere ca fiecare program sa ramina in memorie (sau fie inlocuite de alte programe) si sa nu modifice spatiul de memorie al celorlalte programe.

Programele sub Windows utilizeaza o interfata grafica specifica; se definesc ferestre de program in cadrul carora se desfasoara executia programului.

Comunicarea intre sistemul de operare (inclusive driverele de I/O) se realizeaza prin intermediul mesajelor. Orice program scris sub Windows contine o bucla de testare a aparitiei mesajelor; orice eveniment de intrare/iesire este convertit ( de catre Windows prin intermediul driverelor I/O) intr-un mesaj ce poate fi tratat in program.

Structura generica a unui program in Windows este prezentata in continuare.

Componentele unui program sub Windows sint urmatoarele :

- antetul *windows.h* ( care contine toate tipurile de date, functii si clase Windows)
- functia *WinMain* – echivalentul functiei *main* sub DOS
- o bucla de mesaje Windows
- o functie *callback* de mesaje. Functiile de tip *callback* sint functii carora li se transmite adresa altei functii care revine cu informatii. Functiile callback sint necesare pentru ca programul va utiliza interfata pentru programarea aplicatiilor (API).
- optional un fisier resursa – pentru definirea meniurilor prestabilite de mediul Windows.

Un exemplu simplu de program in Windows este prezentat in continuare.

Acest program defineste si creaza o fereasta de program, o afiseaza pe ecran dupa care se intra in bucla de mesaje care vor fi interpretate.

```
#include <windows.h>
#include "generic.h"

HINSTANCE hInst;          // current instance
LPCTSTR lpszAppName = "Generic";
LPCTSTR lpszTitle = "Generic Application";

int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR
lpCmdLine, int nCmdShow)
{
    MSG msg;
    HWND hWnd;
    WNDCLASS wc;

    wc.style          = CS_HREDRAW | CS_VREDRAW;
    wc.lpfnWndProc    = (WNDPROC)WndProc; // asocierea intre fereasta
//de program si functia de prelucrare a mesajelor primite
    wc.cbClsExtra     = 0;
```

```

wc.cbWndExtra      = 0;
wc.hInstance      = 0;
wc.hIcon          = LoadIcon(hInstance, lpszAppName);
wc.hCursor        = LoadCursor(NULL, IDC_ARROW);
wc.hbrBackground  = (HBRUSH)(COLOR_WINDOW+1);
wc.lpszMenuName    = lpszAppName;
wc.lpszClassName  = lpszAppName;

if(!RegisterClass(&wc))
    return FALSE;
hInst = hInstance;
hWnd = CreateWindow (lpszAppName,
                    lpszTitle,
                    WS_OVERLAPPEDWINDOW,
                    CW_USEDEFAULT, 0,
                    CW_USEDEFAULT, 0,
                    NULL,
                    NULL,
                    hInstance,
                    NULL
                    );

if(!hWnd)
    return FALSE;
ShowWindow(hWnd, nCmdShow);
UpdateWindow(hWnd);
while(GetMessage(&msg, NULL, 0,0))
{
    TranslateMessage(&msg);
    DispatchMessage(&msg);
}
return(msg.wParam);
}

LRESULT CALLBACK WndProc( HWND hWnd, UINT uMsg, WPARAM wParam, LPARAM
lParam)
{
    switch(uMsg)
    {
        case WM_COMMAND:
            switch(LOWORD(wParam))
            {
                case IDM_TEST :
                    break;
                case IDM_EXIT :
                    DestroyWindow(hWnd);
                    break;
            }
            break;

        case WM_DESTROY :
            PostQuitMessage(0);
            break;
        default:
            return (DefWindowProc(hWnd, uMsg, wParam, lParam));
    }
    return(0L);
}

```

Fisierul de resurse asociat programului de mai sus este urmatorul:

```
#include "windows.h"
#include "generic.h"

GENERIC MENU DISCARDABLE
BEGIN
    POPUP "&File"
    BEGIN
        MENUITEM "&Test!",          IDM_TEST
        MENUITEM "E&xit",          IDM_EXIT
    END
    POPUP "&Help"
    BEGIN
        MENUITEM "&About MyApp...",  IDM_ABOUT
    END
END
```

Acest fisier defineste un meniu asociat ferestrei de program si genereaza mesaje (IDM\_TEST, IDM\_EXIT) care vor putea fi tratate in program.

Tratarea oricarui eveniment ce apare in sistemul de calcul se face prin prelucrarea mesajelor asociate acestui eveniment.

Mesajele primite de la tastatura, mouse si timer sint ( detalii in Win32 Reference):

Keyboard Input Messages	Mouse Input Messages	Timer Messages
WM_ACTIVATE WM_CHAR WM_DEADCHAR WM_GETHOTKEY WM_HOTKEY WM_KEYDOWN WM_KEYUP WM_KILLFOCUS WM_SETFOCUS WM_SETHOTKEY WM_SYSCHAR WM_SYSDEADCHAR WM_SYSKEYDOWN WM_SYSKEYUP	WM_CAPTURECHANGED WM_LBUTTONDOWN WM_LBUTTONUP WM_MBUTTONDOWN WM_MBUTTONUP WM_MOUSEACTIVATE WM_MOUSEMOVE WM_MOUSEWHEEL WM_NCHITTEST WM_NCLBUTTONDOWN WM_NCLBUTTONUP WM_NCMBUTTONDOWN WM_NCMBUTTONUP WM_NCMOUSEMOVE WM_NCRBUTTONDOWN WM_NCRBUTTONUP WM_RBUTTONDOWN WM_RBUTTONUP	WM_TIMER