

Cuprins - a doua jumătate

- I. Programare Windows
- II. Structura unei aplicații Windows
- III. Lucrul cu resurse
- IV. Afișarea în Windows
- V. Fișiere în Windows
- VI. Controlul execuției

I. Programare Windows

Ce este Windows?

- sistem de operare
- pe programator îl interesează
 - resursele cu care lucrează
 - tipurile de date
 - cum se armonizează cu tipurile din limbajele de programare
 - funcțiile disponibile pentru accesul la resursele sistemului
 - modul de funcționare al aplicațiilor

Win32API (1)

- interfața de programare oferită de sistemele Windows
- componente
 - controlul interfeței grafice (GDI)
 - gestiunea proceselor
 - comunicarea interproces
 - etc.

Win32API (2)

- implementare în sistemul Windows - 3 biblioteci partajate
 - user32.dll
 - gdi32.dll
 - kernel32.dll
- definesc și manipulează cele 3 categorii de obiecte Windows

Obiecte

- toate resursele sistemului sunt obiecte
- categorii
 - obiecte utilizator
 - ex.: ferestre, meniuri, icon-uri etc.
 - obiecte GDI
 - ex.: fonturi, imagini (*bitmaps*) etc.
 - obiecte nucleu
 - ex.: procese, fire de execuție, semafoare, fișiere etc.

Accesul la obiecte

- nu se poate face direct
 - doar într-un mediu controlat
- aplicația trebuie să obțină un identificador (*handle*) al obiectului dorit
- cu ajutorul acestuia îl poate accesa
- un *handle* este de fapt un indice într-un tabel al sistemului

Tipuri specifice Windows (1)

- HANDLE
 - identificador pentru orice tip de obiect Windows
- identificatori specifici diverselor obiecte
 - HBITMAP - imagini (bitmaps)
 - HFILE - fișiere
 - HMENU - meniuri
 - HWND - ferestre
 - etc.

Tipuri specifice Windows (2)

- tipuri numerice
 - dimensiunile lor sunt fixe - nu depind de compilator
- exemple
 - CHAR, INT, INT32, SHORT, LONG, LONG32 - întregi cu semn
 - BYTE, WORD, DWORD, UCHAR, UINT, USHORT, ULONG - întregi fără semn

Tipuri specifice Windows (3)

- pointeri
- exemple
 - PHANDLE, LPHANDLE
 - PINT, LPINT
 - PSTR, LPSTR - șiruri de caractere
 - etc.
- prefixe: P - pointer, LP - long pointer
 - în practică sunt echivalente

II. Structura unei aplicații Windows

Programare bazată pe evenimente

- *event-driven programming*
- aplicația reacționează la diverse evenimente apărute în sistem
- exemple de evenimente
 - apăsarea unui buton din interfață
 - apăsarea unei taste
 - acționarea mouse-ului

Mesaje

- evenimentele sunt detectate de sistemul Windows
- la apariția unui eveniment, sistemul Windows trimite un mesaj aplicației corespunzătoare
- fiecare aplicație are o buclă principală de mesaje
 - în care preia mesajele primite și le tratează

Ferestre

- o fereastră este elementul principal de interfață
- o aplicație trebuie să aibă cel puțin o fereastră
 - poate avea și mai multe
- tratarea mesajelor se face de fapt la nivelul fiecărei ferestre

Controale

- elemente care pot fi inserate într-o fereastră
- pot genera evenimente
- exemple
 - butoane
 - liste
 - texte
 - statice
 - editabile

Funcții *callback*

- funcția de tratare a mesajelor este scrisă de utilizator
 - numită și funcție a ferestrei
 - este apelată din bucla de mesaje
 - dar nu explicit în codul pe care îl scriem
- este o funcție *callback*
 - trebuie înregistrată în sistem
 - apoi Windows o apelează implicit când e cazul

Structura aplicației Windows (1)

- definirea stilului ferestrei principale a aplicației
 - se folosește o structură `WNDCLASSEX`
- înregistrarea tipului de fereastră definit anterior
- crearea ferestrei pe baza stilului definit
- afișarea ferestrei
- bucla principală de mesaje

Structura aplicației Windows (2)

- în afara funcției principale (*WinMain*) mai trebuie scrisă și funcția de tratare a mesajelor
- dacă aplicația are mai multe ferestre, poate fi necesară repetarea acestor pași pentru fiecare fereastră

Scrierea unei aplicații Windows

- trebuie inclus fișierul `windows.h`, care conține:
- prototipurile funcțiilor Win32API
- definițiile tipurilor de date Windows
- definițiile denumirilor asociate mesajelor Windows
 - ex.: `WM_LBUTTONDOWN` - apăsarea butonului stânga al mouse-ului