

V. Fișiere în Windows

Fișiere

- putem utiliza în continuare bibliotecile de nivel înalt
 - `stdio.h`
 - `iostream.h`
- la nivel jos, acestea se bazează pe funcțiile Windows pentru lucrul cu fișiere
 - tipul utilizat - `HANDLE`
 - în loc de `FILE` sau `fstream`

Operații cu fișiere

- deschidere/creare
 - `CreateFile`
- închidere
 - `CloseHandle`
- citire/scriere
 - `ReadFile`, `WriteFile`
- căutare după nume
 - `FindFirstFile`, `FindNextFile`, `FindClose`

Deschidere/creare

```
HANDLE CreateFile(LPCTSTR  
    lpFileName, DWORD  
    dwDesiredAccess, DWORD  
    dwShareMode,  
    LPSECURITY_ATTRIBUTES  
    lpSecurityAttributes, DWORD  
    dwCreationDisposition, DWORD  
    dwFlagsAndAttributes, HANDLE  
    hTemplateFile);
```

Parametri - valori uzuale (1)

- `lpFileName`
 - numele fișierului
- `dwDesiredAccess`
 - `GENERIC_READ`, `GENERIC_WRITE`
- `dwShareMode`
 - `FILE_SHARE_READ`, `FILE_SHARE_WRITE`
- `lpSecurityAttributes`
 - `NULL`

Parametri - valori uzuale (2)

- `dwCreationDisposition`
 - `OPEN_EXISTING`, `CREATE_ALWAYS`,
`CREATE_NEW`, `TRUNCATE_EXISTING`
- `dwFlagsAndAttributes`
 - `FILE_ATTRIBUTE_NORMAL`
- `hTemplateFile`
 - `NULL`

Citire

```
BOOL ReadFile(HANDLE hFile,  
              LPVOID lpBuffer,  
              DWORD nNumberOfBytesToRead,  
              LPDWORD lpNumberOfBytesRead,  
              LPOVERLAPPED lpOverlapped  
              ) ;
```

Scriere

```
BOOL WriteFile(  
    HANDLE hFile,  
    LPCVOID lpBuffer,  
    DWORD nNumberOfBytesToWrite,  
    LPDWORD  
    lpNumberOfBytesWritten,  
    LPOVERLAPPED lpOverlapped  
);
```


Căutare după nume

```
HANDLE FindFirstFile(LPCTSTR  
    lpFileName, LPWIN32_FIND_DATA  
    lpFindFileData);  
  
BOOL FindNextFile(HANDLE  
    hFindFile, LPWIN32_FIND_DATA  
    lpFindFileData);  
  
BOOL FindClose(HANDLE  
    hFindFile);
```

VI. Controlul execuției

Planificare

- există 3 tipuri de obiecte Windows pentru controlul execuției
 - procese (*processes*)
 - fire de execuție (*threads*)
 - fibre (*fibers*)
- fiecare are funcțiile sale specifice

Procese (1)

- un proces poate crea alte procese (fii)
 - care la rândul lor pot crea alte procese ș.a.m.d.
- la lansare, un program constă dintr-un singur proces
- procesele au spații de memorie complet disjuncte
 - cod, date, stivă
 - dar pot comunica între ele

Procese (2)

- creare
 - `CreateProcess`
- terminare
 - `ExitProcess`
- așteptarea terminării unui proces fiu de către procesul părinte
 - `WaitForSingleObject`

Fire de execuție (1)

- inițial, un proces constă dintr-un singur fir
 - care poate crea alte fire ș.a.m.d.
- firele de execuție partajează mare parte din spațiul de memorie al procesului
 - codul și datele
 - dar nu și stiva
- nucleul este cel care planifică firele pentru execuție la procesor

Fire de execuție (2)

- creare
 - `CreateThread`
 - este apelată o funcție precizată ca parametru pentru `CreateThread`
 - la terminarea acestei funcții, firul de execuție este terminat automat
- așteptarea terminării unui fir de către părintele său
 - `WaitForSingleObject`

Fibre

- asemănătoare firelor de execuție
- planificarea lor este realizată explicit prin program
- creare
 - `CreateFiber`, `ConvertThreadToFiber`
- planificare
 - `SwitchToFiber`