**Programare Procedurala**

**Laborator 10**

**Fisiere**

În stdio.h: tipul **FILE**

Se declară un pointer la **FILE:**

**FILE \* f;**

Secventă tipică de lucru: se deschide, se prelucrează, se închide fişierul.

Fişiere standard predefinite (deschise automat la rularea programului):

* **stdin**: fişierul standard de intrare (normal: tastatura);
* **stdout**: fişierul standard de ieşire (normal: ecranul);
* **stderr**: fişierul standard de eroare (normal: ecranul);

(sunt constante de tipul FILE \* declarate în stdio.h)

Obs: E bine ca mesajele de eroare să fie scrise la stderr, pentru a putea fi separate (prin redirectare) de mesajele normale de ieşire.

**Tip flux de date**

* modul text: doar pt. fişiere cu caractere tipăribile obişnuite , \t, \n
* modul binar: pt. toate celelalte situaţii (chiar şi pt. fişiere text) (asigură corespondenţa exactă între

conţinutul scris şi citit).

**Deschiderea fişierelor**

**FILE \*fopen (const char \*path, const char \*mode);**

* **const char \*path**: numele fişierului (absolut sau față de directorul curent)
* **const char \*mode**: modul de deschidere; primul caracter semnifică:

|  |  |  |  |
| --- | --- | --- | --- |
| **Operație** | **Tip flux** | **Mod acces** | **Observații** |
| citire | text | r / rt | fişierul trebuie să existe. |
| binar | rb |  |
| scriere | text | w / wt | dacă fişierul nu există, e creat;  dacă există, e trunchiat la 0. |
| binar | wb |
| adăugare la sfârṣitul fișierului | text | a / at | dacă fişierul nu există, e creat; |
| binar | ab |  |
| actualizare (citire ṣi scriere cu acelaṣi pointer) | text | r+ / r+t / rt+ | fişierul trebuie să existe. |
| binar | r+b / rb+ |  |
| actualizare cu înlocuirea fiṣierului existent | text | w+ / w+t /wt+ |  |
| binary | w+b / wb+ |  |
| actualizare la sfârṣitul fiṣierului | text | a+ / a+t / at+ |  |
| binar | a+b / ab+ |  |

* Returnează pointer la fişier sau **NULL** dacă fişierul nu poate fi deschis; valoarea returnată este xmemorată în variabila fişier, care a fost declarată pentru accesarea lui.

*Exemplu:*

**FILE \*f1,\*f2;**

**char \*name = "fisier2.txt";;**

**f1 = fopen("fisier1.txt", "wt");** //va deschide fisierul **fisier1.txt** pentru scriere text;

**if (!(f2 = fopen(name, "rt+")))** { /\* trateazã eroarea \*/ }

**else** { /\* lucreazã cu fişierul \*/ }

**Închiderea fişierelor**

**int fclose(FILE \*stream);**

* închide fişierul şi eliberează zona tampon;
* returnează 0 în caz de succes, **EOF** în caz de eroare

**Citire/scriere (fișiere text)**

**int fgetc(FILE \*f);** /\* citeşte caracter din fișier, **EOF** la sfârşit de fişier \*/

**int fputc(int c, FILE \*f); /\***scrie caracter în fişier \*/

**int ungetc(int c, FILE \*f);** // pune caracterul **c** în bufferul asociat lui **f** (**c** va fi următorul caracter citit din **f**)

**char\* fgets(char \*s, int n, FILE \*f);**

/\* - citeşte până la (inclusiv) linie nouã, sau max. n-1 caractere;

* adaugă '\0' la sfârşit;
* deci face citirea sigură a unei linii, fără depăşire;
* returnează NULL dacă apare EOF înainte de a fi citit ceva \*/

**int fprintf(FILE \*f, const char \*format, ...);**

**int fscanf(FILE \*f, const char \*format, ...);**

#### Citire/scriere fără conversie (fișere binare)

**size\_t fread(void \*ptr, size\_t size, size\_t nrec, FILE \*f);**

**size\_t fwrite(const void \*ptr, size\_t size, size\_t nrec, FILE \*f);**