

## 1 Manipulation de fichiers en C

Tout **fichier** est manipulé via un descripteur de type **FILE\***.

Les fonctions de manipulation de fichiers nécessitent **#include <stdio.h>**.

**Ouverture** : **FILE\*** fopen (**char** \*filename, **char** \*mode);

- **filename** : chaîne représentant le chemin du fichier.
- **mode** : chaîne spécifiant si ouvert en lecture et/ou écriture.
  - "r" : (read) ouverture en lecture (fichier déjà existant)
  - "w" : (write) ouverture en écriture (destruction si existant)
  - "a" : (append) écriture, rajoute à la fin, (création si non existant)
  - "r+", "w+", "a+" : variantes (c.f. **man 3 fopen**).
  - Suffixe "b" : (binary) lire/écrire les données sans transformation.
- Succès si le pointeur retourné  $\neq$  NULL.

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

- **stream** : pointeur sur le descripteur du fichier.
- Succès si la valeur retournée = 0.

Fermeture **importante** : garantit l'**écriture effective** des données.

Lectures et écritures «avancent» **automatiquement** dans le fichier.

**Lecture formatée** : **int** fscanf (FILE \*stream, **char** \*format, ...) ;

⇒ comme scanf avec le **descripteur** en premier argument.

**Écriture formatée** : **int** fprintf (FILE \*stream, **char** \*format, ...) ;

⇒ comme printf avec le **descripteur** en premier argument.

**Lecture par blocs** : **size\_t** fread (**void** \*ptr, **size\_t** size, **size\_t** nitems, FILE \*stream);

**Écriture par blocs** : **size\_t** fwrite (**void** \*ptr, **size\_t** size, **size\_t** nitems, FILE \*stream);

Test de fin de fichier atteinte : **int** feof (FILE \*stream); Retour :  $\neq$  0 si fin de fichier atteinte suite aux lectures **précédentes**.

⇒ Toujours avoir **lu avant de tester** avec feof.

Connaître la **position actuelle** (lect/écr) dans un fichier : **long** ftell (FILE \*stream);

Se déplacer (sans lire) dans un fichier : **int** fseek (FILE \*stream, **long** offset, **int** whence);