



## Fiche : FILE – Fichiers en C (lib. standard)

### 1 – Ouverture de fichiers

Prototype :

```
FILE *fopen(const char *filename, const char *mode);
```

Description générale :

Cette fonction réalise l'ouverture du fichier. Si ce fichier n'existe pas, il est créé.

Les paramètres d'entrées :

filename : Pointeur sur la chaîne contenant le nom du fichier à ouvrir.

mode : Pointeur sur un caractère indiquant le mode d'ouverture du fichier.

Valeurs possibles pour mode :

- "r" : Ouverture pour lecture seule
- "w" : Création du fichier et écriture (si fichier pré-existant, il est écrasé)
- "a" : Mise à jour par écriture en fin de fichier
- "r+" : Ouverture en lecture et en écriture
- "w+" : Idem w mais avec possibilité de lire
- "a+" : Idem a mais avec possibilité de lire

Valeurs de retour :

Un pointeur sur une structure de type FILE à utiliser dans toutes les autres fonctions d'accès aux fichiers.

Si l'ouverture s'est mal passée, la fonction retourne NULL.

Remarque :

Lorsqu'un fichier est ouvert dans un mode permettant l'écriture et la lecture il est interdit de passer directement d'une lecture à une écriture ou d'une écriture à une lecture. Entre les deux il faut repositionner le pointeur sur le fichier en appelant la fonction fseek.

### 2 – Fin de fichier atteinte en lecture

Prototype :

```
int *feof(FILE * stream);
```

Description générale :

Cette fonction teste si la fin de fichier est atteinte.

Les paramètres d'entrées :

stream : le pointeur sur la structure FILE associée à un fichier lors de l'ouverture

Valeurs de retour :

Si une lecture a été effectuée APRES le dernier octet du fichier, alors cette fonction retourne une valeur différente de zéro.

### 3 – Fermeture de fichier

Prototype :

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

Description générale :

Fermeture du fichier identifié par son pointeur sur la structure FILE.

Le paramètre d'entrée :

stream : le pointeur sur la structure FILE associée à un fichier lors de l'ouverture

Valeurs de retour :

0 si la fermeture s'est bien passée, sinon EOF



---

## 4 – Déplacement dans un fichier

---

Prototype :

```
int fseek(FILE *stream, long offset, int whence);
```

Description générale :

Permet de se déplacer dans un fichier précédemment ouvert.

Les paramètres d'entrée :

stream : le pointeur sur la structure FILE associée à un fichier lors de l'ouverture.

whence : indique une position particulière. 3 constantes sont possibles :

SEEK\_SET : Début du fichier

SEEK\_END : Fin du fichier

SEEK\_CUR : A partir de la position actuelle

offset : valeur représentant le nombre d'octet à ajouter à la position particulière pour obtenir un déplacement.

Exemples :

Si Offset=2 et whence= SEEK\_SET, l'appel de la fonction fseek positionne le curseur 2 octets après le début du fichier.

Si Offset=0 et whence= SEEK\_CUR, l'appel de la fonction fseek ne déplace pas le curseur. **Par contre, cela autorise une lecture suite à une écriture ou vice-versa.**

Valeurs de retour :

0 en cas de succès, sinon une valeur différente de 0

---

## 5 – Indication de position dans le fichier

---

Prototype :

```
long int ftell(FILE *stream);
```

Description générale :

Retourne le nombre d'octets existants entre le début du fichier et la position courante.

Le paramètre d'entrée :

stream : le pointeur sur la structure FILE associée à un fichier lors de l'ouverture.

Valeurs de retour :

Le nombre d'octets existants entre le début du fichier et la position courante.

Remarque :

Cette fonction, associée à fseek, permet de déterminer la longueur d'un fichier.