# Manipulez des fichiers avec C

## 1. Objectifs

## 2. Introduction

- La manipulation de fichiers en **C** implique l'utilisation de fonctions standard de la bibliothèque `stdio.h` pour **ouvrir**, **lire**, **écrire** et **fermer** des fichiers.
- En **C**, les fichiers sont gérés à l'aide de fonctions standard de la bibliothèque **stdio.h**. Les principales opérations sont l'**ouverture**, la **lecture**, l'**écriture** et la **fermeture** de fichiers.

## 1. Création d'une application FX test sous Eclipse.

### 1. Ouvrir un fichier

- Pour ouvrir un fichier, utilisez la fonction fopen. Elle prend deux arguments : le nom du fichier et le mode d'ouverture. Voici quelques modes courants :
  - **`"r"`**: ouvrir un fichier en lecture.
  - **`"w"`**: ouvrir un fichier en écriture (crée un fichier vide ou écrase le contenu existant).
  - **`"a"`**: ouvrir un fichier en ajout (append) (écrit à la fin du fichier).
  - **`"r+"`**: ouvrir un fichier en lecture et écriture.
  - **`"w+"`**: ouvrir un fichier en lecture et écriture (crée un fichier vide ou écrase le contenu existant).
  - **`"a+"`**: ouvrir un fichier en lecture et ajout (append).

- **Exemple de lecture et d'écriture dans un fichier**
  - Voici un exemple complet montrant comment lire d'un fichier et écrire dans un autre fichier en utilisant C.

```
#include <stdio.h>
#include <stdlib.h>

int main(void) {
```

apcpedagogie.com

```
    FILE *sourceFile, *destFile;
    char ch;

    // Ouvrir le fichier source en lecture
    sourceFile = fopen("source.txt", "r");
    if (sourceFile == NULL) {
        perror("Erreur d'ouverture du fichier source");
        return 1;
    }

    // Ouvrir le fichier de destination en écriture
    destFile = fopen("destination.txt", "w");
    if (destFile == NULL) {
        perror("Erreur d'ouverture du fichier de destination");
        fclose(sourceFile);  // Fermer le fichier source avant de quitter
        return 1;
    }

    // Lire du fichier source et écrire dans le fichier de destination
    while ((ch = fgetc(sourceFile)) != EOF) {
        if (fputc(ch, destFile) == EOF) {
            perror("Erreur d'écriture dans le fichier de destination");
            fclose(sourceFile);
            fclose(destFile);
            return 1;
        }
    }

    // Fermer les fichiers
    fclose(sourceFile);
    fclose(destFile);

    printf("Copie terminée avec succès !\n");
    return 0;
}
```

## 1. Réalisation

1. **Inclusion de bibliothèques :**

apcpedagogie.com

```
#include <stdlib.h>   // Pour les opérations d'entrée/sortie standard
#include <stdlib.h>  // Pour les fonctions d'utilitaires, comme exit
```

1. **Déclaration des variables de fichiers :**

```
FILE *sourceFile, *destFile;  // Pointeurs de type FILE pour les fichiers source et destination
char ch;  // Variable pour stocker chaque caractère lu
```

1. **Ouverture des fichiers :**

```
sourceFile = fopen("source.txt", "r"); ouvre source.txt en mode lecture.
destFile = fopen("destination.txt", "w"); ouvre destination.txt en mode écriture.
```

- Chaque ouverture est suivie d'une vérification pour s'assurer que l'opération a réussi. En cas d'échec, un message d'erreur est affiché et le programme se termine.

- **Lecture et écriture :**
  - La boucle

```
while ((ch = fgetc(sourceFile)) != EOF)
```

  - lit chaque caractère du fichier source jusqu'à la fin du fichier (EOF). fputc(ch, destFile); écrit le caractère lu dans le fichier de destination. Chaque écriture est vérifiée pour détecter d'éventuelles erreurs.

- **Fermeture des fichiers :**

```
fclose(sourceFile); et fclose(destFile);
```

  - ferment les fichiers ouverts pour libérer les ressources.

- **Message de confirmation :**

apcpedagogie.com

```
printf("Copie terminée avec succès !\n");
```

- affiche un message indiquant que l'opération de copie est terminée avec succès.
- *Autres fonctions utiles*
    - **`fprintf et fscanf :`**
    - Utilisées pour écrire et lire des données formatées dans un fichier, respectivement.

```
fprintf(filePointer, "Format string", var1, var2, ...);
fscanf(filePointer, "Format string", &var1, &var2, ...);
```

    - **`fgets et fputs :`**
    - Utilisées pour lire et écrire des chaînes de caractères.

```
fgets(buffer, size, filePointer);  // Lire une ligne du fichier
fputs(buffer, filePointer);        // Écrire une chaîne dans le fichier
```

    - Ces fonctions fournissent une flexibilité pour différentes opérations de lecture et d'écriture dans des fichiers en C

---

**Cours en C**

Source : https://apcpedagogie.com/manipulez-des-fichiers-avec-c · téléchargé le 07/11/2026 · © APC Pédagogie · apcpedagogie.com

apcpedagogie.com