

Langage C

Lecture et écriture dans des fichiers texte (ASCII)



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

Les fichiers

Un programme a souvent besoin :

- ❖ de lire des données (texte, nombres, images, mesures ...)
- ❖ de sauvegarder des données (texte, nombres, images, mesures ...)

Cela se fait en lisant et en écrivant dans des fichiers.

Deux types de fichiers existent :

- ❖ les fichiers de données “brutes” écrits en binaire, non “lisibles” directement par l’humain.
- ❖ les fichiers de données écrits au format texte (appelé aussi format ASCII), directement lisibles par l’humain.

Le type FILE

- ❖ Pour manipuler un fichier, on utilise un **pointeur** sur une donnée spécifique dont le type est **FILE** (structure prédéfinie que nous n'avons pas besoin de connaître précisément) :

FILE* **fichier ;**

- ❖ La variable **fichier** contiendra l'adresse en mémoire du début du fichier. C'est le **nom "interne" du fichier**.

Ouverture de fichiers

fopen est définie dans le fichier `stdio.h` par :

```
FILE* fopen (char *nom, char *mode)
```

- ❖ **nom** est une chaîne de caractères (tableau de caractères) contenant le nom du fichier ou bien un flot de données standard (stdin, ...)
- ❖ **mode** désigne le type de traitement des données :

“**r**” (read) : lecture (si le fichier existe)

“**w**” (write) : écriture (le fichier est écrasé s’il existe et s’il n’existe pas, il est créé)

“**a**” (append) : écriture à la fin d’un fichier existant

Ouverture de fichiers

Où se trouve le fichier ouvert ?

```
fichier = fopen("essai.txt", "r");
```

Dans ce cas, le fichier `essai.txt` est dans le répertoire du projet de C

Comment travailler sur un fichier situé ailleurs ?

Le chemin absolu d'accès au fichier peut être donné in extenso (attention les `\` de windows deviennent des `/` en C)

```
fichier = fopen("C:/Data/fichier1.txt", "r");
```

A quoi sert l'extension `.txt` ?

A rien pour le Langage C.

Elle permet en revanche à l'OS de l'ordinateur de sélectionner le programme permettant d'ouvrir le fichier.

Fermeture de fichiers

Il faut toujours fermer un fichier après l'avoir utilisé afin de libérer la mémoire et de provoquer le transfert des données.

Fermeture d'un fichier à l'aide de la fonction `fclose` :

```
fclose(fichier) ;
```

`fclose` est définie dans le fichier `stdio.h` par :

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

Lecture/écriture dans des fichiers texte

Écriture: `fprintf(FILE *fichier, char *proto, ...);`

exemple: format données à
écrire

```
double a;  
fprintf(fichier, "%lf", a);
```

Tout se passe comme si on envoyait la donnée a dans un fichier au lieu de l'envoyer à l'écran.

Lecture: `fscanf(FILE *fichier, char *proto, ...);`

exemple:

```
double a;  
fscanf(fichier, "%lf", &a);
```

Tout se passe comme si on lisait la donnée a dans un fichier au lieu de la lire au clavier.

Écriture dans un fichier

```
#include <stdio.h>

int main()
{
    double a=1.5, b=2.5;
    FILE* fichier;
    // Ouverture du fichier en écriture grâce à "w"
    fichier = fopen("essai.txt", "w");
    // Vérification de l'ouverture du fichier
    if (fichier != NULL)
    {
        fprintf(fichier, "%lf\n", a);    // Écriture
        fprintf(fichier, "%lf\n", b);    // Écriture

        fclose(fichier);                // Fermeture du fichier
    }
    return 0;
}
```

Lecture dans un fichier

```
#include <stdio.h>
int main()
{
    int i;
    double tab[2];
    FILE* fichier;
    // Ouverture du fichier en lecture grâce à "r"
    fichier = fopen("essai.txt", "r");
    // Vérification de l'ouverture du fichier
    if (fichier != NULL)
    {
        for(i=0; i<2; i++)
            fscanf(fichier, "%lf", &tab[i]); // Lecture
        fclose(fichier); // Fermeture du fichier
    }
    for(i=0; i<2; i++)
        printf("%lf\n", tab[i]); //vérification : on affiche 1.5 puis 2.5
    return 0;
}
```