



Les fichiers
Dorian.H Mekni
09 | Mar | 2020

fichier

- Les fichiers sont classés et organisés sur leur support suivant un système de fichiers.
- C'est ce dernier qui permet à l'utilisateur d'ordonner ses fichiers dans une arborescence de dossiers.
- Cela facilite leur localisation et gestion.

fopen() | fclose()

- La creation du fichier s'initialise au préalable dans son dossier au sein de son ordinateur.
- Une fois dans la console ou bien sur l'IDE, le fichier s'ouvre au sein d'une fonction, ici en l'occurrence ce sera le int main(void)
- Chaque fichier s'ouvre en faisant appel à la fonction fopen()
- Il se ferme également en faisant appel à la fonction fclose()

les fichiers en action

```
12 int main(void)
13 {
14     FILE *fic = fopen("sauvegarde.txt", "r");
15     int letter = 0;
16
17     if(fic == NULL)
18         exit(1);
19
20     while(1)
21     {
22         letter = fgetc(fic);
23
24         iffeof(fic))
25             break;
26
27         printf("%c\n ", letter);
28     }
29
30     fclose(fic);
31
32     return 0;
33
34 }
```

fopen() - 1

- La valeur retournée par `fopen` est un flot de données.
- Si l'exécution de cette fonction ne se déroule pas normalement, la valeur retournée est le pointeur `NULL`.
- Il est donc préférable de tester à chaque fois si la valeur renvoyée par la fonction `fopen` est égale à `NULL` dans le but de diagnostiquer des erreurs

fopen() - 2

- *fopen(<nom du fichier>, <mode>)*
- Le premier argument est le nom du fichier concerné qui se manifeste sous une chaîne de caractères.
- Il est préférable de définir le nom du fichier via une CONSTANCE avec la directive #define
- Le deuxième argument spécifie le mode d'accès au fichier.

Les modes d'accès (1)

- Les objets de spécifications des modes varient selon le type de fichier:
- -> fichiers textes: les caractères de contrôle type \n seront lu ainsi
- -> fichiers binaires: les caractères de contrôle ne sont pas interprétés

Les modes d'accès (2)

"r"	ouverture d'un fichier texte en lecture
"w"	ouverture d'un fichier texte en écriture
"a"	ouverture d'un fichier texte en écriture à la fin
"rb"	ouverture d'un fichier binaire en lecture
"wb"	ouverture d'un fichier binaire en écriture
"ab"	ouverture d'un fichier binaire en écriture à la fin
"r+"	ouverture d'un fichier texte en lecture/écriture
"w+"	ouverture d'un fichier texte en lecture/écriture
"a+"	ouverture d'un fichier texte en lecture/écriture à la fin
"r+b"	ouverture d'un fichier binaire en lecture/écriture
"w+b"	ouverture d'un fichier binaire en lecture/écriture
"a+b"	ouverture d'un fichier binaire en lecture/écriture à la fin

Les modes d'accès (3)

- mode contient r -> le fichier existe
- mode contient a-> le fichier n'existe pas
- mode contient w -> le fichier n'existe pas

Les 3 flows

- 3 flots standards permettent la lecture d'un fichier sans que l'on fasse appel à `fopen()` ou à `fclose()`
- -> `stdin` [standard input] — via le clavier -par default-
- -> `stdout` [standard output] — via l'écran -par default-
- -> `stderr` [standard error] — via l'écran -par default-

fclose()

- fclose() permet la fermeture du flot engendré par l'initialisation du dossier ouvert par fopen()
- Le flot est celui du Type *FILE
- Si l'opération s'est bien déroulé il y a un retour 0
- En revanche en cas d'erreur le renvoie d'une valeur non NULL nous est renvoyé.

fprintf()

- Fonction analogue à printf()
- Elle permet l'écriture d'une chaîne de caractères au sein d'un fichier
- Elle s'écrit sur le même modèle que printf()

fscanf()

- Fonction analogue à scanf()
- Elle rend accessible la lecture de données
- Elle s'écrit sur le meme modele que scanf()

fgetc()

- Lecture et écriture d'un fichier
- Type int
- Retourne le caractère lu qui est contenu dans le fichier
- Meme similarity que getchar(), il faut préciser le type int pour capter la fin du fichier.

fputc()

- Elle permet l'écriture du caractère dans le flot de donnée
- Ce dernier est également de type int (un entier)
- L'entier qui correspond au caractère se voit retourné par cette fonction

ungetc()

- Cette fonction permet de remplacer un caractère de type int dans un flot
- ungetc() position le caractère qui s'est converti en unsigned char() dans le flot.
- ungetc() peut être utilise par n'importe quel caractère sauf eof.

Entrée - Sortie binaires

- Elles possèdent un avantage considerable comparé aux fonctions d'entree et de sortie standard.
- En effet elles permettent les entrees comme les sorties sans transcodage.
- Plus adequate pour manipuler des fichiers de grande taille ou de type composé
- `fread()` lit les données du flot et `fwrite()` les écrit.

Le positionnement dans un fichier

- Les fonctions entree-sortie permettent d'accéder à un fichier de manière séquentielle.
- Il est possible aussi d'accéder de manière directe à n'importe quel endroit du fichier, on utilise pour cela la fonction `fseek()`.

Les 3 fonctions entrée | sortie

- SEEK_SET [valeur 0] -> début de fichier
- SEEK_CUR[valeur 1] -> position courante
- SEEK_END[valeur 2] -> fin du fichier
- chacune de ses fonctions retourne la position courante dans le fichier en nombre exact d'octets.

```

/*
LES BASES | . fopen(<fichier>, <mode_ouverture>) : r (lecture seule, doit exister)
                                           : w (écriture)
                                           : a (ajout fin de fichier)
                                           : r+ (lecture | écriture, doit exister)
                                           : w+ (lecture | écriture, fin d'utilisation fichier)
                                           : a+ (ajout lecture | écriture, fin d'utilisation fichier)
. fclose(<fichier>) : fermer un fichier ouvert tout comme ce que l'on faisait une fois que l'on
    avait fini l'utiliser la fonction malloc()
. feof(<fichier>) : Pour évaluer la fin d'un fichier

LECTURE | . fgetc() : permet de lire un caractère.
. fgets(<chaine>, <taille_chaine>, <fichier>) : Elle permet la lecture d'un texte, 1 ligne ou
    une chaîne de caractère. Elle renvoie NULL.
. fscanf(): permet la lecture de texte formaté à partir d'1 fichier.

ECRITURE | . fputc(<caractère>, <format>, ...) : écriture d'1 caractère
. fputs(<chaine>, <fichier>) : écriture d'1 texte, une ligne
. fprintf(<fichier>, <format>, ...) : écriture d'1 texte formaté

POSITION | . ftell(<fichier>) : renvoie à la position du curseur au sein du fichier
. fseek(<fichier>, <deplacement>) : déplace curseur
-> <origine> : SEEK_SET (le début du fichier)
              SEEK_CUR (la position courante)
              SEEK_END (la fin du fichier)
. rewind(<fichier>) : réinitialisation du positionnement curseur

AUTRES | . rename(<ancien nom>, <nouveau nom>)
. remove(<fichier>)

*/

```

