

La gestion des fichiers

MP2I - Informatique

Anthony Lick

Lycée Janson de Sailly

Fichiers

Le C offre la possibilité de **lire** et d'**écrire** des données dans un **fichier**.

Pour des raisons d'efficacité, les accès à un fichier se font par l'intermédiaire d'une **mémoire-tampon (buffer)**, ce qui permet de réduire le nombre d'accès aux périphériques (disque dur, ...).

Fichiers

Pour pouvoir manipuler un fichier, un programme a besoin d'un certain nombre d'informations : l'adresse de l'endroit de la mémoire-tampon où se trouve le fichier, la position de la tête de lecture, le mode d'accès au fichier (lecture ou écriture),...

Ces informations sont rassemblées dans une structure dont le type, **FILE ***, est défini dans la librairie **stdio.h**.

Un objet de type **FILE *** est appelé **flot de données** (**stream** en anglais).

Fichiers

Avant de lire ou d'écrire dans un fichier, on notifie son accès par la commande **fopen**.

Cette fonction prend comme argument le nom du fichier, négocie avec le système d'exploitation et initialise un flot de données, qui sera ensuite utilisé lors de l'écriture ou de la lecture.

Après les traitements, on annule la liaison entre le fichier et le flot de données grâce à la fonction **fclose**.

Ouverture et fermeture d'un fichier

La fonction fopen

1

```
fopen("nom de fichier", "mode")
```

fopen

La fonction **fopen**, de type **FILE***, ouvre un fichier et lui associe un flot de données.

La valeur renvoyée par **fopen** est un flot de données.

Si l'exécution ne se déroule pas normalement, la valeur renvoyée est le pointeur **NULL**.

Il est donc recommandé de toujours tester si la valeur renvoyée par la fonction **fopen** est égale à **NULL** afin de détecter les erreurs (lecture de fichier inexistant, accès interdit, ...).

La fonction fopen

1

```
fopen("nom de fichier", "mode")
```

fopen

Le premier argument de **fopen** est le nom du fichier concerné, fourni sous forme d'une chaîne de caractères.

On préférera définir le nom du fichier dans une constante afin de rendre notre code plus modulaire.

Le second argument, **"mode"**, est une chaîne de caractère qui spécifie le mode d'accès au fichier.

La fonction fopen

"r"	ouverture en lecture
"w"	ouverture en écriture
"a"	ouverture en écriture à la fin
"r+" ou "w+"	ouverture en lecture et en écriture
"a+"	ouverture en lecture et en écriture à la fin

Modes d'accès

- Les modes "r" et "r+" ne fonctionnent que si le fichier existe déjà.
- Les modes "w" et "w+" créent le fichier s'il n'existe pas, et **effacent** le contenu du fichier s'il existe déjà.
- Les modes "a" et "a+" créent le fichier s'il n'existe pas, et rajouteront les données **à la fin** du fichier s'il existe déjà.

La fonction fopen

Flots standard

Trois **flots standard** peuvent être utilisés en C sans qu'il soit nécessaire de les ouvrir ou de les fermer :

- **stdin** (standard input) : unité d'entrée (par défaut, le clavier);
- **stdout** (standard output) : unité de sortie (par défaut, l'écran);
- **stderr** (standard error) : unité d'affichage des messages d'erreur (par défaut, l'écran);

Attention

Il est fortement conseillé d'afficher systématiquement les messages d'erreur sur **stderr** afin que ces messages apparaissent à l'écran même lorsque la sortie standard est redirigée.

La fonction `fclose`

1

```
fclose(flott)
```

fclose

La fonction **fclose** permet de fermer le flot qui a été associé à un fichier par la fonction **fopen**.

Sa syntaxe est celle ci-dessus, où **flott** est le flot de type **FILE*** retourné par la fonction **fopen** correspondant.

La fonction **fclose** retourne un entier qui vaut zéro si l'opération s'est déroulée normalement (et une valeur non nulle en cas d'erreur).

Les entrées/sorties formatées

La fonction d'écriture fprintf

1 `fprintf(flout, "chaîne de contrôle", expression1, ..., expressionN)`

fprintf

La fonction **f**printf, analogue à **printf**, permet d'écrire des données dans un fichier.

Sa syntaxe est celle ci-dessus, ou **flout** est le flot de données (de type **FILE***) retourné par la fonction **fopen**.

Les autres arguments suivent les mêmes règles que pour la fonction **printf**.

La fonction de lecture fscanf

1 `fscanf(flout, "chaîne de contrôle", expression1, ..., expressionN)`

fscanf

La fonction **fscanf**, analogue à **scanf**, permet de lire le prochain mot dans un fichier.

Sa syntaxe est celle ci-dessus, ou **flout** est le flot de données (de type **FILE***) retourné par la fonction **fopen**.

Les autres arguments suivent les mêmes règles que pour la fonction **scanf**.

Elle renvoie la valeur **EOF (end of file)** lorsqu'on arrive à la fin du fichier.

La fonction `rewind`

```
1 int rewind(FILE *fplot);
```

rewind

La fonction **rewind** (**rembobiner** en anglais) permet de revenir au début d'un fichier.

Son **prototype** est celui ci-dessus.

Exemple

exemple.c (1/3)

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  void ajout_eleve(char *fichier)
6  {
7      char nom[255];
8      int date;
9      printf("Nouvel élève :\n");
10     printf("\tNom : ");
11     scanf("%s", nom);
12     printf("\tDate de naissance (JJMAA) : ");
13     scanf("%d", &date);
14
15     FILE *flot = fopen(fichier, "a");
16     if (flot != NULL)
17     {
18         fprintf(flout, "nom: %s\n", nom);
19         fprintf(flout, "date: %d\n", date);
20         fclose(flout);
21     }
22 }
```

Exemple

exemple.c (2/3)

```
1 void cherche_eleve(char *fichier, char *nom)
2 {
3     char buff[255];
4     FILE *flot = fopen(fichier, "r");
5     if (flot != NULL)
6     {
7         while (fscanf(flott, "%s", buff) != EOF)
8         {
9             if (strcmp(buff, nom) == 0)
10            {
11                fscanf(flott, "%s", buff); // buff contient "date:"
12                int date;
13                fscanf(flott, "%d", &date); // le mot scanné est une date au format JJMMAA
14                printf("Elève trouvé :\n");
15                printf("\tNom : %s\n", nom);
16                printf("\tDate de naissance : %d\n", date);
17
18                fclose(flott);
19                return; // Interrompt la fonction
20            }
21        }
22        fprintf(stderr, "Elève inconnu : %s\n", nom);
23        fclose(flott);
24    }
25 }
```

Exemple

exemple.c (3/3)

```
1  int main()  
2  {  
3      char *fichier = "eleves.txt";  
4      for (int i = 0; i < 3; i++)  
5      {  
6          ajout_eleve(fichier);  
7      }  
8      char nom[255];  
9      printf("Entrer un nom d'élève : ");  
10     scanf("%s", nom);  
11     cherche_eleve(fichier, nom);  
12     return EXIT_SUCCESS;  
13 }
```

Output

```
Nouvel élève :  
    Nom :Janson  
    Date de naissance (JJMAA) :010203  
Nouvel élève :  
    Nom :Sally  
    Date de naissance (JJMAA) :112233  
Nouvel élève :  
    Nom :Bob  
    Date de naissance (JJMAA) :000000  
Entrer un nom d'élève : Sally  
Elève trouvé :  
    Nom : Sally  
    Date de naissance : 112233
```

Output

```
Nouvel élève :  
    Nom :Janson  
    Date de naissance (JJMAA) :010203  
Nouvel élève :  
    Nom :Sally  
    Date de naissance (JJMAA) :112233  
Nouvel élève :  
    Nom :Bob  
    Date de naissance (JJMAA) :000000  
Entrer un nom d'élève : John  
Elève inconnu : John
```