

Input/Output en C

Afin de lire ou écrire dans des fichiers en C, on utilise les fonctions de la librairie `stdio.h`, notamment les fonctions suivantes :

- `void fprintf(FILE* fp, ...)` qui se comporte comme `printf` sauf qu'elle prend un argument supplémentaire : un descripteur `FILE*` `fp` du fichier dans lequel elle écrit ;
- `void fscanf(FILE* fp, ...)` qui se comporte comme `scanf` sauf qu'elle prend un argument supplémentaire : un descripteur `FILE*` `fp` du fichier dans lequel elle doit lire.

Le descripteur de fichiers (de type `FILE*` `fp`) sont obtenus grâce à la fonction `fopen`. On ouvre un fichier nommé `file.txt` :

- en mode lecture par l'appel `FILE* fp = fopen("file.txt", "r");` ("r" pour *read*);
- en mode écriture par l'appel `FILE* fp = fopen("file.txt", "w");` ("w" pour *write*);
- en mode ajout à la fin par l'appel `FILE* fp = fopen("file.txt", "a");`;

Si le fichier `file.txt` n'existe pas encore, l'ouvrir en mode lecture renvoie le pointeur `NULL`, tandis qu'en mode écriture ou ajout il sera créé. Si au contraire il existe déjà, l'ouvrir en mode écriture avec "r" écrasera son contenu. Un fichier ouvert en lecture ne peut qu'être lu, un fichier ouvert en écriture ne peut être qu'écrit. Après usage, on ferme un fichier ouvert de descripteur `fp` par l'appel `close(fp)`.

Par exemple, le programme ci-dessous exécuté en présence d'un fichier `a.txt` de la forme :

```
toto
3 4
```

produit un fichier `b.txt` de la forme :

```
(toto)4 3
```

```
1 #include<stdio.h>
2 #include<stdlib.h>
3
4 int main() {
5     FILE* fp1 = fopen("a.txt", "r");
6     FILE* fp2 = fopen("b.txt", "w");
7     char buffer[100];
8     int tmp1;
9     int tmp2;
10    fscanf(fp1, "%s\n", buffer);
11    fprintf(fp2, "(%s)", buffer);
12    fscanf(fp1, "%d ", &tmp1);
13    fscanf(fp1, "%d", &tmp2);
14    fprintf(fp2, "%d ", tmp2);
15    fprintf(fp2, "%d", tmp1);
16    fclose(fp1);
17    fclose(fp2);
18    return 0;
19 }
```

Le comportement de `scanf` avec le spécificateur de format `%s` obéissant à des règles difficiles à deviner, il est conseillé de faire des tests. À titre d'exemple, si la première ligne du fichier `a.txt` est changée en " toto" le résultat du programme donné en exemple est le même, mais avec la ligne " toto . tutu" on obtient des 0 au lieu de 3 et 4...