



Fiche : Lecture-Ecriture – Fichier texte en C

1 – Fonctions gérant un caractère

Prototypes :

```
int fgetc(FILE *stream);  
int fputc(int c, FILE *stream);
```

Description générale :

fgetc permet de lire / obtenir un caractère depuis un fichier précédemment ouvert.

fputc permet d'écrire / placer un caractère dans un fichier précédemment ouvert.

Ces deux fonctions sont donc réciproques.

Les paramètres d'entrées :

stream : pointeur sur une structure de type FILE initialisée précédemment par la fonction fopen

c : le caractère à écrire dans le fichier

Valeurs de retour :

fgetc renvoie le caractère lu dans le fichier, Sinon, EOF en cas d'erreur telle que la fin de fichier.

fputc. renvoie EOF en cas d'erreur, sinon le caractère écrit dans le fichier.

2 – Fonctions gérant des chaînes

Prototype :

```
char *fgets(char *s, int n, FILE *stream);  
int fputs(const char *s, FILE *stream);
```

Description générale :

fgets permet de lire / obtenir une chaîne depuis un fichier précédemment ouvert.

fputs permet d'écrire / placer une chaîne dans un fichier précédemment ouvert.

Ces deux fonctions sont donc réciproques.

Les paramètres d'entrées :

s : pointeur sur la chaîne à écrire dans le fichier (fputs) ou destinée à recevoir la copie de la chaîne lue dans le fichier (fgets).

stream : pointeur sur une structure de type FILE initialisée précédemment par la fonction fopen

n (Pour fgets) : indique le nombre de place dispos pour les données hors caractère terminateur.

La lecture d'une chaîne dans le fichier s'arrêtera lorsque :

- Fin de fichier
- Buffer s plein
- Fin de chaîne (saut de ligne détecté)

Valeurs de retour :

fgets renvoie NULL en cas d'erreur (fin de fichier), Sinon, le pointeur sur la chaîne.

fputs renvoie EOF en cas d'erreur, sinon une valeur positive.



3 – Fonctions gérant des valeurs de type divers

Prototype :

```
int fprintf(FILE *stream, const char *format[, arg, ...]);  
int fscanf(FILE *stream, const char *format[, address, ...]);
```

Description générale :

fprintf permet d'écrire sous forme de texte des variables de type divers et variés.

fscanf permet de lire sous forme de texte des variables.

Ces deux fonctions sont donc réciproques.

Les paramètres d'entrée :

stream : pointeur sur une structure de type FILE initialisée précédemment par la fonction fopen

format : chaîne de caractère définissant le format d'affichage de chacune des variables

arg : liste en nombre non limité de variables à écrire dans le fichier

address : liste en nombre non limité de pointeurs de variables destinées à recevoir le résultat de la lecture dans le fichier.

Valeurs de retour :

fscanf retourne le nombre de champs lus dans le fichier

fprintf retourne le nombre de caractères écrit dans le fichier (terminateur non écrit et donc non compté)

Exemples :

Soit 4 variables à écrire dans un fichier :

```
int A=10 ;  
char B='A' ;  
float C=5.15 ;  
char Chaine[10]="Bonjour" ;  
FILE * pFichier ;  
...  
... // Ouverture du fichier  
// Sauvegarde  
fprintf (pFichier, "%d,%c,%d,%02x,%2.2f,%s\n", A, B, B, B, C, Chaine) ;  
fclose(pFichier) ;
```

Le fichier contient alors :

10, A, 65, 41, 5.15, Bonjour + Changement de ligne

La ligne suivant permet de relire dans le fichier les variables précédentes :

(nouvelles variables de type char nommées D et E)

```
fscanf (pFichier, "%d,%c,%d,%02x,%2.2f,%s\n", &A, &B, &D, &E, &C, Chaine) ;
```