

FICHIERS texte

Un fichier texte est un fichier qui contient du texte ASCII.

On appelle lecture dans un fichier le transfert de données du fichier vers la mémoire centrale. La lecture dans un fichier texte est analogue à la lecture au clavier : le texte vient du fichier au lieu de venir du clavier.

On appelle écriture dans un fichier le transfert de données de la mémoire centrale vers le fichier. L'écriture dans un fichier texte est analogue à l'écriture à l'écran : le texte va dans le fichier au lieu de s'afficher.

Ouverture et fermeture d'un fichier texte :

Pour lire ou écrire dans un fichier texte, on a besoin d'un pointeur de fichier.

Le pointeur de fichier nous permet de désigner le fichier dans lequel nous souhaitons Lire ou écrire. Un pointeur de fichier est de type FILE *. On déclare un tel pointeur comme toute autre variable :

```
FILE *f ; /* déclaration d'un pointeur de fichier f */
```

Avant de pouvoir écrire ou lire dans le fichier, il faut lier le pointeur de fichier à un fichier sur le disque. On appelle cette opération l'ouverture de fichier. L'ouverture se fait avec la fonction fopen, qui prend en paramètre le nom du fichier sur le disque.

Ex : FILE *f ;

```
f = fopen("fichier.txt", "rt") ;
```

Le premier paramètre est le nom du fichier, ou plus exactement le chemin vers le fichier.

Le deuxième paramètre est le mode d'ouverture, et "rt" signifie "fichier texte ouvert en lecture seule.

La fonction fopen retourne le pointeur NULL, en cas d'erreur d'ouverture de fichier.

Après avoir utilisé un fichier, il faut le refermer en utilisant fclose.

La fonction `fclose` prend en paramètre le pointeur de fichier et ferme le fichier.

Lecture et écriture des données formatées :

Pour lire des données dans un fichier texte, on utilise la fonction `fscanf`, qui est analogue à `scanf` sauf qu'elle lit dans un fichier au lieu de lire au clavier. La fonction `fscanf` prend pour premier paramètre le pointeur de fichier, puis les autres paramètres sont les mêmes que ceux de `scanf`.

La fonction `fscanf` retourne le nombre de variables effectivement lues.

Ex : `FILE *f ;`

```
int x ;  
.....  
fscanf(f,"%d",&x) ;
```

Exemple :

Transfert les éléments de fichier dans un tableau.

```
#include <stdio.h>  
#include <stdlib.h>  
#define NB_ELEM_MAX 100  
int chargefichier(int tab[NB_ELEM_MAX])  
{FILE *f ;  
int i=0 ;  
f=fopen("fichier.txt", "rt") ;  
if (f==NULL)  
{puts("Erreur d'ouverture de fichier : inexistant") ;  
exit(1) ;  
}  
while (i < NB_ELEM_MAX && fscanf(f, "%d",&tab[i])==1)  
i++ ;  
fclose(f) ;  
return i ;  
}  
void affiche(int tab[], int n)  
{int i ;  
for (i=0 ; i<n ; i++)  
printf("tab[%d] = %d\n",i, tab[i]) ;  
}
```

```

main()
{int tab[NB_ELEM_MAX];
  int n ;
  n=chargefichier(tab) ;
  affiche(tab,n) ;
}

```

Ecriture des données formatées :

Pour écrire des données dans un fichier texte, on peut utiliser la fonction `fprintf`, qui est analogue à la fonction `printf`. La fonction `fprintf` prend comme paramètre un pointeur de fichier. Les autres paramètres sont les mêmes que pour `printf` ; le second paramètre est la chaîne de format avec le texte à écrire et les `%d`, `%f`, puis les variables à écrire séparées par des virgules.

Exemple :

Écrire un programme qui lit un fichier texte contenant des nombres entiers, et le copie dans un deuxième fichier texte où chaque nombre est multiplié par 3. En cas d'erreur la fonction retourne 1 sinon 0 est retourné au programme appelant.

```

#include<stdio.h>
int triple( )
{FILE *fr,*fw ;
  int n ;
  fr = fopen("fichieralire.txt", "rt") ;
  fw=fopen("fichieraecrire.txt", "wt") ;
  if (fr==NULL || fw==NULL)
    return 1 ;
  while (fscanf(fr, "%d", &n)==1)
    fprintf(fw, "%d ", 3*n) ;
  fclose(fr) ;
  fclose(fw) ;
  return 0 ;
}
int main ( )
{int err ;
  err = triple() ;
  if (err != 0)
    puts("Erreur d'ouverture de fichier ") ;
  return 0 ;
}

```