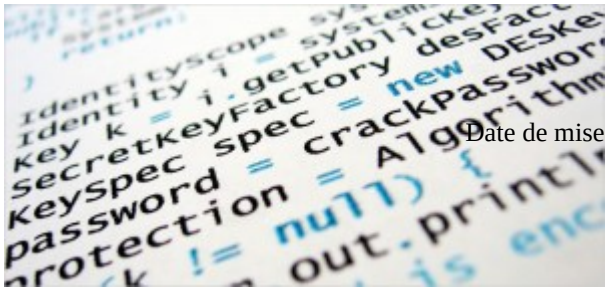
Fiche 2 : Manipulation de Fichiers

- Développement - C & C++ -



Date de mise en ligne : mardi 19 novembre 2013

Description :

Le but de cet article n'est pas de faire un cours sur les instructions permettant de faire des opérations sur les fichiers mais simplement de donner un bref rappel sur leur création et leur manipulation.

Attention : il est nécessaire d'avoir déjà acquis les bases de la programmation en C afin d'apprécier au mieux ces exemples.

Sommaire :

I) Ouvrir un fichier

Haut de page

Avant d'ouvrir un fichier, il faut déclarer une variable de type *File* via l'instruction suivante :

```
FILE* fichierAOuvrir = NULL;
```

On notera ici qu'on utilise un pointeur pour créer la variable.

Ensuite l'ouverture d'un fichier se fait en utilisant la fonction *fopen* :

```
int main (void)
{
 char fileName[] = "monfichier.txt";
 FILE *fichierAOuvrir = NULL;

 fichierAOuvrir = fopen(fileName , "w");
 return 0;
}
```

Ici, "w" permet l'ouverture du fichier en écriture. Si le fichier n'existe pas, il sera créé.

II) Vérifier l'existence d'un fichier

Haut de page

Pour vérifier l'existence d'un fichier avant de l'ouvrir, il faut mettre en place une condition avec un *if* :

```
int main(void)
{
 char fileName[] = "monFichier.txt";

 if (fileName != NULL)
 {
 fichierAOuvrir = fopen(fileName , "w");
 }
 else
 {
 printf("Erreur : le fichier %s n'existe pas !", fileName);
```

```
 }
 return 0;
```

} III) Ecrire des données dans un fichier

Haut de page

Pour écrire dans un fichier, il y a deux fonctions principales :

- *fputs* qui permet d'écrire une chaîne de caractère simple.
- *fprintf* qui permet d'écrire une chaîne de caractère composée.

Mais pour expliquer tout ça, mettons nous en situation :
fputs s'utilise de cette manière...

```
fputs("chaine de caractère à écrire", fichierAOuvrir);
```

...et *fprintf* comme ceci :

```
int num = 2;
fprintf(fichierAOuvrir, "chaine de caractère numéro %d", num);
```

Ici, on écrit une chaîne de caractères formatée avec l'utilisation de la variable num.

IV) Lire des données dans un fichier

Haut de page

Pour lire des données, là encore il y a deux possibilités (principales) :

- *fgets* qui permet de récupérer une ligne dans le fichier
- *fscanf* qui permet de récupérer des informations formatées dans un ordre bien précis

Pour expliquer ces fonctions, voici les situations suivantes (n'hésitez pas à consulter le chapitre consacré au débogage si vous avez des problèmes) :

- La fonction *fgets*

La fonction *fgets* s'utilise en suivant ce modèle (on suppose que le fichier existe et qu'il est ouvert) :

```
char chaine[] = NULL;
printf("Affichage du contenu du fichier.txt en utilisant fgets");
while(fgets(chaine, 1000, fichierAOuvir) != NULL)
{
 printf("%s", chaine);
}
fclose(fichierAOuvrir);
```

Ici, on affichera des lignes pouvant atteindre jusqu'à 1000 caractères.
Etant donné que nous n'avons plus besoin de lire dans ce fichier, on le ferme avec la fonction fclose(nom de la variable FILE) ;

- La fonction *fscanf*

Pour décrire l'utilisation de la fonction *fscanf*, il faut tout d'abord créer un second fichier dans lequel vous allez saisir les données suivantes :

```
A 1 B 2
```

Ensuite on peut passer à la programmation :

```
char car[2] = NULL;
int entier[2] = 0;
fileName = "fichier2.txt"; // mettre le nom du deuxième fichier
fichierAOuvrir = fopen(fileName, "r"); //ouverture en lecture seule
printf("Affichage du contenu du fichier %s avec la fonction fscanf :\n", fileName);
fscanf(fichierAOuvrir, "%c %d %c %d", &amp;amp;amp;amp;amp;amp;car[0], &amp;amp;amp;amp;amp;amp;entier[0],
&amp;amp;amp;amp;amp;amp;car[1], &amp;amp;amp;amp;amp;amp;entier[1])
printf("informations : %c %d %c %d\n\n", car[0], entier[0], car[1], entier[1]);
fclose(fichierAOuvrir);
```

V) Débogage

Haut de page

En suivant le même paramétrage d'ouverture de fichier que dans le chapitre 2, alors vous allez être confronté à un problème si vous utilisez la fonction *fgets* directement après le *fprintf*.

En effet, le fichier a été ouvert en écriture seul, ce qui veut dire qu'on ne peut pas le lire !

Afin de réaliser cette tâche, il faut d'abord fermer le fichier puis le rouvrir en lecture seul, c'est à dire, avec l'option *"r"* en lieu et place de *"w"*.

Une manière plus simple de traiter ce problème est d'ouvrir le fichier une première fois avec l'option *"r+"*, ce qui vous permettra de le lire ET écrire dedans.

VI) Conclusion

Haut de page

Dans cette article nous avons pu voir comment ouvrir un fichier, le lire, y écrire, et le fermer.
Il existe d'autre manière d'ouvrir un fichier. On peut, par exemple, faire en sorte de remplacer le texte existant par une nouvelle chaîne de caractères ou ouvrir le fichier pour écrire après la dernière ligne (options *"a+"* ou *"w+"*).
De même qu'il existe la possibilité de lire ou d'écrire seulement un caractère dans un fichier (fonctions *fgetc* et *fputc*).