**Les fichiers**

1. D**éfinition et déclaration**

Un fichier est une structure informatique permettant le stockage des données d’une façon permanente sur un support informatique externe à la mémoire centrale (disque dur, clé usb, etc).

Un fichier peut contenir des données de deux types :

* le type texte : toutes les données sont représentées sous forme de suite de caractères. Le nombre 12.5 est écrit par exemple : ‘1’, ‘2’, ‘.’, ‘5’. Ces fichiers sont éditables et modifiables avec un simple éditeur de texte.
* Le type binaire : les données sont stockées selon leur format de représentation en mémoire centrale. Donc le nombre 12.5 est écrit sous forme d’une suite binaire (0,1) sous 4 octets. Exemple : les fichiers exécutables, images, audios etc.

L’accès aux données stockées dans un fichier peut se faire selon deux types :

* l’accès séquentiel : on accède à une données en faisant une lecture de toutes les données se trouvant avant celle-ci.
* l’accès direct : Pour accéder à une donnée cible, il suffit de réaliser un déplacement de lecture directement sur celle-ci. Le déplacement se calcule par la taille en octets des données à lire.

Déclaration d’un fichier :

* Algo : nomFichier : fichier
* Langage C : FILE \* nomFichier ; //pointeur de fichier
* PHP Pas de déclaration.
1. Opérations sur fichier

1. Ouverture du fichier

Avant toute utilisation d’un fichier, il faut l’ouvrir en différents modes : Lecture, écriture (avec écrasement), ajout (avec conservation).

| Algorithme |
| --- |
| nomFichier <- ouvrir(nomPhysique, mode)mode : lecture, écriture, ajoutnomPhysique : c’est le nom donné par le système d’exploitation avec une extension : mesCours.txt  |

| Langages C et PHP |
| --- |
| nomFichier = fopen (nomPhysique, mode) ;mode : **r** (read), **w** (write), **a** (append) |

1. Fermeture du fichier

Tout fichier ouvert doit être fermé après utilisation.

| Algorithme  | Langages C et PHP  |
| --- | --- |
| fermer (nomFichier) | fclose (nomFichier) |

1. Lecture et écriture

En C et PHP, il y a plusieurs fonctions de lecture et écriture.

| Algorithme  | Langages C et PHP  |
| --- | --- |
| lire (nomFichier, nomVariable)  | fgetc : lire un caractère (char)fgets : lire une chaine (string)fread : lecture binaire fscanf : lecture formatée  |
| écrire (nomFichier, nomVariable)  | fputc : écrire un caractère (char)fputs : écrire une chaine (string)fwrite : écriture binaire fprintf : écriture formatée  |

1. Test de fin de fichier

La fin d’un fichier est matérialisée par un caractère (nom imprimable) : EOF. End Of File.

| Algorithme  | Langages C et PHP  |
| --- | --- |
| fin\_fichier (nomFichier)  | feof (nomFichier) |

Cette fonction permet de tester la fin d’un fichier, elle retourne un booléen (vrai ou faux).

3. Exemple de lecture d’un fichier

Ecrire un algo, un programme C et un développement PHP qui permet de demander à un utilisateur un **nom** de fichier et affiche son contenu à l’écran. La lecture se fera **caractère** par caractère.

Algo : affiche\_fichier

Déclaration

 f : fichier

 nom : chaine

 c : caractère

Début

 Afficher (“Donner le nom de votre fichier :”)

 saisir (nom)

 f <- ouvrir (nom, “lecture”)

 Tant que fin\_fichier (f) = faux faire

 lire (f , c)

 afficher ( c )

 Fin tant que

 fermer (f)

fin affiche\_fichier

Traduction en C :

#include <stdio.h>

int main (){

 FILE \* f ;

 char nom [30] ;

 char c ;

 printf("Donner le nom de votre fichier :");

 scanf("%s", &nom);

 f = fopen (nom, "r");

 while ( ! feof (f)){

 c = fgetc (f);

 printf("%c", c );

 }

 fclose (f) ;

 return 0;

}

Développement en PHP :

<!DOCTYPE html>

<html>

<head>

 <title>Affiche fichier</title>

</head>

<body>

<center>

 <h1> Programme qui affiche un fichier</h1>

 <form method="post">

 Nom du fichier : <br/>

 <input type="text" name="nom"> <br/>

 <input type="submit" name="Afficher" value="Afficher">

 </form>

 <?php

 if (isset($\_POST["Afficher"])){

 $nom = $\_POST["nom"];

 $f = fopen ($nom, "r");

 while ( ! feof ($f)){

 $c = fgetc ($f);

 echo $c;

 }

 fclose ($f) ;

 }

 ?>

</center>

</body>

</html>

**Exercice 1 :** Ecrire un programme en C et PHP qui permet de simuler le copier / coller d’un système d’exploitation entre un fichier source contenant un texte et un fichier destination nouveau.

#include <stdio.h>

int main (){

 FILE \* s, \*d ;

 char nomS [30], nomD[30] ;

 char c ;

 printf("Donner le nom de votre fichier source :");

 scanf("%s", &nomS);

 printf("Donner le nom de votre fichier destination :");

 scanf("%s", &nomD);

 s = fopen (nomS, "r");

 d = fopen (nomD, "w");

 while ( ! feof (s)){

 c = fgetc (s);

 fputc (c, d);

 }

 fclose (s) ;

 fclose (d);

 printf( "La recopie est terminee ");

 return 0;

}

<!DOCTYPE html>

<html>

<head>

 <title>Copier-Coller</title>

</head>

<body>

<center>

 <h1> Programme Copier - Coller</h1>

 <form method="post">

 Nom du fichier source : <br/>

 <input type="text" name="nomS"> <br/>

 Nom du fichier destination : <br/>

 <input type="text" name="nomD"> <br/>

 <input type="submit" name="Valider" value="Valider">

 </form>

 <?php

 if (isset($\_POST["Valider"])){

 $nomS = $\_POST["nomS"];

 $nomD = $\_POST["nomD"];

 $s = fopen ($nomS, "r");

 $d = fopen ($nomD, "w");

 while ( ! feof ($s)){

 $c = fgetc ($s);

 fputs ($d, $c);

 }

 fclose ($s) ;

 fclose ($d);

 echo " <br> La recopie est terminee ";

 }

 ?>

</center>

</body>

</html>

Exercice 2 : utilisation de fprintf

Ecrire un programme qui permet de saisir les données des produits (désignation, prix, quantité) et d’éditer une facture dans un fichier texte. Utiliser la fonction fprintf pour écrire les données dans le fichier. La facture est composée de plusieurs produits.

#include <stdio.h>

int main (){

FILE \* fact ;

char designation [30] ;

float prix;

int qte , i , nbProduits ;

fact = fopen (“facture.txt”, “w”);

printf(“Donner le nombre de produits :”);

scanf(“%d”, &nbProduits);

for (i = 1; i<= nbProduits ; i++){

 printf(“Donner la designation :”);

scanf(“%s”, &designation);

printf(“Donner le prix :”);

scanf(“%f”, &prix);

printf(“Donner la quantite:”);

scanf(“%d”, &qte);

fprintf(fact, “%s %f %d \n”, designation, prix, qte );

}

fclose (fact);

printf(“Enresgitrement de la facture dans le fichier”);

return 0;

}