

# PHP – Master CCI

Bertrand Estellon

Aix-Marseille Université

January 30, 2014

# Ouverture d'un fichier

```
<?
$id = fopen("toto.txt", "w");
var_dump($id); /* 1 */
fclose($id);
var_dump($id); /* 2 */

$id = tmpfile();
var_dump($id); /* 3 */
fclose($id);
var_dump($id); /* 4 */
?>
```

- Une bonne façon de faire :

```
<?
$id = fopen("toto.txt", "w");
if ($id===FALSE) die("Erreur");
$r = fclose($id);
if ($r===FALSE) die("Erreur");
?>
```

- 1 resource(5) of type (stream)
- 2 resource(5) of type (Unknown)
- 3 resource(6) of type (stream)
- 4 resource(6) of type (Unknown)

# Ouverture d'un fichier

- r** : ouverture en lecture seule et la lecture commence au début du fichier. *Le fichier doit exister.*
- r+** : le fichier est ouvert en lecture et écriture, les opérations commencent au début. *Le fichier doit exister.*
- w** : le fichier est ouvert en écriture et l'écriture commence au début du fichier. *Le fichier est créé s'il n'existe pas.*
- w+** : le fichier est ouvert en écriture et en lecture et les opérations commencent au début du fichier. *Le fichier est créé s'il n'existe pas.*
- a** : le fichier est ouvert en écriture et l'écriture commence à la fin du fichier. *Le fichier est créé s'il n'existe pas.*
- a+** : le fichier est ouvert en écriture et lecture. L'écriture commence à la fin et la lecture au début. *Le fichier est créé s'il n'existe pas.*

# Verrouillage de fichiers

## ► Problème de concurrence :

- Plusieurs clients demandent une page simultanément
- ⇒ deux scripts modifient un fichier en même temps
- ⇒ Conflit

## ► Solution : verrouillage du fichier

```
<?
```

```
$id = fopen("toto.txt", "w");  
echo $id."\n";  
flock($id, LOCK_EX); // Verrouillage en lecture et en écriture  
....  
flock($id, LOCK_UN); // Déverrouillage  
fclose($id);
```

```
?>
```

## ► LOCK\_SH : Verrouillage en écriture seulement

# Écriture dans un fichier

&lt;?

```
$id = fopen("toto.txt", "w");  
flock($id, LOCK_SH);  
fwrite($id, "mon texte\n"); // Écriture de "mon texte$\backslashbacksla  
$nb = 100;  
fwrite($id, $nb); // Écriture de "100"  
flock($id, LOCK_UN);  
fclose($id);
```

?&gt;

- Contenu du fichier toto.txt à la fin de l'exécution :

mon texte  
100

# Exemple d'écriture dans un fichier

```
<html>
<body>
  <form action="<? echo $_SERVER['PHP_MYSELF']; ?>" method="post">
    <b>Nom :</b> <input type="text" name="nom"/><br/>
    <b>Prenom :</b> <input type="text" name="prenom"/><br/>
    <input type="submit" value="Envoyer"/>
  </form>
  <?
    if (isset($_POST['nom']) && isset($_POST['prenom'])) {
      $nom = $_POST['nom'];
      $prenom = $_POST['prenom'];
      $id = fopen("liste.txt", "a");
      if ($id===FALSE) die("erreur");
      flock($id, LOCK_SH);
      fwrite($id, "$nom;$prenom\n");
      flock($id, LOCK_UN);
      $r = fclose($id);
      if ($r===FALSE) die("erreur");
    }
  <?>
</body>
</html>
```

jean;pascal  
didier;julien  
paul;jacques  
michel;jean

# Lecture dans un fichier

- Contenu du fichier toto.txt avant l'exécution :

mon texte  
100

&lt;?

```
$id = fopen("toto.txt", "r");  
$s = fgets($id, 256);  
var_dump($s); // string(10) "mon texte$\backslashbackslash$n"  
$s = fgets($id, 256);  
var_dump($s); // string(4) "100$\backslashbackslash$n"  
fclose($id);
```

?&gt;

- La fonction fgets prend deux paramètres : une ressource pointant sur un fichier et le nombre maximum de caractères à lire

# Exemple de lecture dans un fichier

```
<table border="1">
<tr><th>Nom</th><th>Prenom</th></tr>
<?
$id = fopen("liste.txt", "r");
while ($ligne=fgets($id)) {
    $t = explode(";", $ligne);
    echo "<tr><td>$t[0]</td><td>$t[1]</td></tr>";
}
fclose($id);
?>
</table>
```

jean;pascal  
didier;julien  
paul;jacques  
michel;jean

⇒

Nom	Prenom
jean	pascal
didier	julien
paul	jacques
michel	jean



# Lecture de données formatées

```
<table border="1">  
  <tr><th>Nom</th><th>Prenom</th></tr>
```

```
<?
```

```
  $id = fopen("liste.txt", "r");  
  while ($t=fgetcsv($id, 100 /* 1 */, ";" /* 2 */)) {  
    echo "<tr><td>$t[0]</td><td>$t[1]</td></tr>";  
  }  
  fclose($id);
```

```
?>
```

```
</table>
```

❶ : nombre maximum de caractères à lire

❷ : caractère séparateur

```
jean;pascal  
didier;julien  
paul;jacques  
michel;jean
```



Nom	Prenom
jean	pascal
didier	julien
paul	jacques
michel	jean

# Lecture de la totalité d'un fichier

```
<table border="1">
  <tr><th>Nom</th><th>Prenom</th></tr>
  <?
    $f = file("liste.txt");
    foreach ($f as $ligne) {
      $t = explode(";", $ligne);
      echo "<tr><td>$t[0]</td><td>$t[1]</td></tr>";
    }
  ?>
</table>
```

```
jean;pascal
didier;julien
paul;jacques
michel;jean
```



Nom	Prenom
jean	pascal
didier	julien
paul	jacques
michel	jean

# Manipulations de fichiers

- Copie de fichiers :

```
<?
$res = copy("liste.txt", "liste2.txt");
if ($res===FALSE) die("erreur");
?>
```

- Renommer un fichier :

```
<?
$res = rename("liste.txt", "liste2.txt");
if ($res===FALSE) die("erreur");
?>
```

- Supprimer un fichier :

```
<?
$res = unlink("liste.txt");
if ($res===FALSE) die("erreur");
?>
```

# Manipulations de fichiers

## ► Existence :

```
<?
if (file_exists("liste.txt"))
    echo "le fichier existe";
else echo "le fichier n'existe pas";
?>
```

## ► Créer un fichier vide :

```
<?
if (!file_exists("liste.txt"))
    touch("liste.txt", time());
?>
```

## ► Taille d'un fichier :

```
<?
$nombre_octets = filesize("liste.txt");
echo $nombre_octets;
?>
```

# Autres fonctions utiles

- ▶ `is_file("fichier") = true` s'il s'agit d'un fichier
- ▶ `is_readable("fichier") = true` si le fichier est dispo en lecture
- ▶ `is_writable("fichier") = true` si le fichier est dispo. en écriture
- ▶ `filetype("fichier") =` le type du fichier
- ▶ `basename("img/truc/toto.php") = "toto.php"`
- ▶ `realpath("toto.php") =` chemin complet du fichier
- ▶ `fseek($id, n) →` se positionne sur le n-ème octet
- ▶ `rewind($id) →` se positionne au début du fichier
- ▶ `ftell($id) =` position courante du curseur dans le fichier
- ▶ `fgetc($id) =` le prochain caractère du fichier