PHP Command Line Interface

D.Moreaux

2 octobre 2023

Table des matières

Introduction

Si il est principalement utilisé dans le contexte du Web, le langage PHP peut aussi être utilisé en ligne de commande pour réaliser des programmes tournant sur la console.

L'utilisation en ligne de commande présente quelques différences par rapport à l'utilisation au travers d'un serveur web :

— durée d'exécution illimitée
— permet de passer des paramètres sur la ligne de commande comme un programme en C (ou autre)
— permet un programme interactif (lecture du clavier)
— ne demande pas de serveur web

1 Installation

1.1 Sous Linux

Sous Linux, il suffit d'installer le package `php-cli`. La commande `php` se trouvera dans le répertoire `/usr/bin` et sera donc accessible directement.

Le fichier de configuration se trouvera dans `/etc`, souvent dans un répertoire `php`.

Il faudra penser à installer les modules désirés et généralement, l'installeur se chargera de les activer automatiquement.

Si on désire qu'un programme s'exécute directement en PHP, il suffit de le faire commencer par une ligne

```
#!/usr/bin/php
```

et de lui donner les droits d'exécution `chmod 755 xxx.php`. A noter que le fichier n'a pas besoin d'avoir l'extension `.php` pour que cela fonctionne.

1.2 MS Windows

L'installation sous MS Windows est un peu plus compliquée :

— Aller à `http://php.net/downloads` et cliquer sur **Windows downloads**.
— Télécharger l'archive `.zip` de la dernière version de PHP **Non Thread Safe** pour la plateforme désirée (32 ou 64 bits).
— Décompresser l'archive dans une sous répertoire (pour l'exemple, nous prendrons `C:\PHP` mais il est possible d'utiliser un autre nom ou un autre répertoire)

— Dans le pannneau de Contrôle, choisir **System** et **Advanced System Settings**. Demander à changer les variables d'environnement et changer le `PATH` système (onglet inférieur) pour ajouter le répertoire d'installation (`C:\PHP`).
— Ouvrir une ligne de commande en mode administrateur et taper `PATH` pour s'assurer que le répertoire se trouve bien dans le texte affiché.
— Taper `php -v` et s'assurer que le numéro de version affiché est bien correct (et que le programme se lance bien)
— Pour associer l'interpréteur PHP aux fichiers PHP, taper les deux commandes suivantes dans la ligne de commande administrateur :
`assoc␣.php=phpfile`
`ftype␣␣phpfile="C:\PHP\php.exe"␣-f␣"%1"␣--␣%~2`
Une fois cela fait, il suffira de double cliquer sur un fichier PHP pour l'exécuter (attention, la fenêtre se ferme automatiquement en fin de programme).

Pour faire des modifications dans le fichier de configuration (par exemple activer des extensions), il faudra copier le fichier php.ini-development sous le nom php.ini dans le répertoire d'installation.

Il faudra ensuite chercher la ligne `extension_dir` en commentaire et y ajouter une ligne

```
extension_dir␣=␣"C:\PHP\ext"
```

Pour activer une extension, chercher la ligne `extension=` correspondant à l'extension à activer et décommenter (en retirant le `;`.

2 Accès aux fichiers

Les accès aux fichiers se feront à l'aide de fonctions similaires aux instructions du langage C.

Exemple :

```
$fh=fopen("monfichier","r");
while␣(($s=fgets($fh))!==␣FALSE)␣{
␣␣echo␣$s;
}
fclose($fh);
```

2.1 fopen

```
$fh=fopen("test.txt","rb");
```

Pour ouvrir un fichier, on utilisera `fopen`. Cette fonction prend deux paramètres semblables à ceux en langage C : le nom du fichier et le mode.

Pour le mode, il est possible d'ajouter un caractère `t` ou `b` à la fin pour spécifier si on manipule un fichier texte ou binaire. En mode texte, une conversion des retours de fin de ligne se fait automatiquement.

Il est cependant conseillé de n'utiliser que le mode binaire pour une meilleure compatibilité entre systèmes.

Les modes possibles sont dans le tableau suivant :

Mode	R/W	position	
r	R	début	
r+	RW	début	
w	W	début	crée ou tronque
w+	RW	début	crée ou tronque
a	W	fin	
a+	RW	fin	
x	W	début	exclusif, erreur si le fichier existe
x+	RW	début	exclusif, erreur si fichier existe
c	W	début	crée

La fonction `fopen` retourne un *file handle* qui sera utilisé par les autres fonctions d'accès aux fichiers.

2.2 fclose

```
fclose($fh);
```

La fonction `fclose` permet de fermer un fichier ouvert par `fopen`. Cela effectuera les écritures qui étaient encore au niveau des buffers et n'auraient pas encore été transcrites sur le disque dur.

2.3 fflush

```
fflush($fh);
```

La fonction `fflush` permet de forcer l'écriture sur le disque dur des données qui sont dans les buffers.

2.4 feof

```
if (!feof($fh)) ...
```

La fonction `feof` retournera TRUE si on est arrivé à la fin du fichier (lors de lectures).

2.5 fseek

```
fseek($fh,$pos);
fseek($fh,$pos,SEEK_SET);
```

La fonction `fseek` permet de positionner le pointeur dans le fichier avant des lectures ou écritures.

Le deuxième paramètre indique la position (positive ou négative) par rapport à la référence indiquée en 3eme paramètre.

La référence sera `SEEK_SET` pour fournir une position par rapport au début du fichier, `SEEK_END` pour une position par rapport à la fin du fichier et `SEEK_CUR` pour une position par rapport à la position courante. Si la référence n'est pas indiquée, la valeur `SEEK_SET` sera utilsée par défaut.

2.6 ftell

```
$pos=ftell($fh);
```

La fonction `ftell` permet de récupérer la position courante dans le fichier.

2.7 fgetc

```
$c=fgetc($fh);
```

La fonction `fgetc` permet de lire un caractère dans le fichier. Si on est arrivé à la fin du fichier, retourne FALSE (tester avec ===).

2.8 fgets

```
$s=fgets($fh);
$s=fgets($fh,$maxlen);
```

La fonction `fgets` permet de lire une chaîne de caractère dans le fichier.

La chaîne s'arrête quand une des conditions suivantes est atteinte : fin de fichier, retour à la ligne (inclus dans la chaîne retournée) ou `$maxlen`-1 caractères lus

Si on est à la fin du fichier, retourne FALSE.

Contrairement à la version du langage C, la version PHP de `fgets` est sûre à l'emploi (pas de risque de buffer overflow).

2.9 fread

```
$s=fread($fh,$len);
```

La fonction `fread` permet de lire `$len` octets dans le fichier. Cette fonction est *binary safe* (elle gère correctement les octets à 0 ou autres valeurs spéciales).

2.10 fwrite

```
fwrite($fh,$s);
```

La fonction `fwrite` permet d'écrire une chaîne dans une fichier. Elle est binary safe.

2.11 file

```
$a=file("test.txt"};
$a=file("test.txt",FILE_IGNORE_NEW_LINES);
```

La fonction `file` permet de lire la totalité d'un fichier dans un tableau. Chaque ligne du fichier sera sauvé comme une ligne dans le tableau de chaînes retourné.

L'option `FILE_IGNORE_NEW_LINES` permet d'éviter que les retours de chariot ne soient inclus dans les chaînes du tableau.

3 Console

3.1 Paramètres

Lorsque le programme est exécuté, les paramètres transmis sont sauvés dans le tableau `$argv` et le nombre de paramètres dans `$argc`. Le nom de l'exécutable se trouve dans le paramètre 0.

Si on désire passer des paramètres qui pourraient être interprétés par le moteur PHP, il suffit de commencer la liste de paramètres par un double tiret `--`

```
php␣test.php␣--␣-a␣123␣-b␣-c
```

3.2 Entrée clavier

Dans un programme en ligne de commande, on peut utiliser les *file handlers* `STDIN`, `STDOUT` et `STDERR` pour accéder à la console. Plus particulièrement, une lecture sur `STDIN` se traduira par une lecture au clavier (en attendant l'appui de Enter)

Il est également possible d'ouvrir `stdin` à l'aide de l'instruction

```
$stdin=fopen("php://stdin","r");
```

3.3 Séquences ANSI

Il est possible d'effacer l'écran ou de positionner le curseur à l'écran à l'aide de séquences de code ANSI. De nombreuses séquences existent et permettent de changer la couleur, de déplacer le curseur ou d'effacer une partie de l'écran.

Effacer l'écran :

```
echo "\033[2J";
```

Positionner à l'écran :

```
echo "\033[".$row.";".$col."H";
```

On peut trouver une liste de codes à

`https://en.wikipedia.org/wiki/ANSI_escape_code#Terminal_output_sequences`

Sous certaines versions de MS Windows, il sera nécessaire d'activer les codes ANSI. Cela peut être fait en créant une entrée dans la base de registre :

Branche : `HKCU\Console` Clé : `VirtualTerminalLevel` Valeur : `0x1`

Attention, cela affectera tous les terminaux ouvert et pourrait poser problème avec certaines applications. Avant de faire cette opération, vérifier si les séquences ANSI sont reconnues.