



Dans ce TP, nous allons apprendre à créer un script Shell.

1. Création du fichier

Un script n'est autre qu'un fichier ordinaire, que nous allons créer avec n'importe quel éditeur de texte, en l'occurrence Vi (ou Nano).

```
$ vim essai.sh
```

Remarque :

« .sh » est une extension optionnelle qui permet de définir un fichier de type script Shell.

2. Indiquer le nom du Shell utilisé par le script

Il existe plusieurs Shell : "sh", "bash", "ksh", etc. Le plus riche et le plus répandu est Bash. Notre script doit commencer en indiquant le Shell utilisé et sa location dans Linux.

Code : Bash

```
#!/bin/bash
```

Remarque :

Nous pouvons remplacer /bin/bash par /bin/sh, /bin/ksh selon la syntaxe du Shell que nous voulons utiliser.

Après cette première instruction, nous pouvons insérer n'importe quelle commande que nous tapions dans l'invite de commande: ls, cd, mkdir, etc.

Voici un script bash qui permet d'afficher le contenu du répertoire en cours:

Code : Bash

```
#!/bin/bash  
ls
```

3. Les commentaires

En Shell, les commentaires commencent par un # :

Code : Bash

```
#!/bin/bash  
# Affichage de la liste des fichiers  
ls
```

Sauvegarder votre fichier et quitter Vim (taper ":wq" ou encore ":x").

4. Rendre le script exécutable

Code : Console

```
$ chmod +x essai.sh
```

Vérifier que le droit a bien été donné :

Code : Console

```
$ ls -l essai.sh  
-rwxr-xr-x 1 info info 17 2019-03-13 14:33 essai.sh
```

5. Exécuter le script

Pour exécuter un script, nous ajoutant « ./ » devant son nom:

Code : Console

```
$ ./essai.sh
```

Il a affiché la liste de tous les fichiers contenus dans le répertoire en cours. Nous allons modifier le script de telle façon à afficher le répertoire en cours:

Code : Bash

```
#!/bin/bash  
pwd  
ls
```

Les commandes seront exécutées l'une après l'autre :

Code : Console

```
$ ./essai.sh  
/home/monpc  
essai.sh
```

6. Débogage du script

Pour déboguer un script. Nous utilisons la commande suivante:

Code : Console

```
$ bash -x essai.sh
```

Le Shell affiche alors le détail de l'exécution de votre script.

Code : Console

```
$ bash -x essai.sh  
+ pwd  
/home/monpc  
+ ls  
essai.sh
```

7. Transformer le script en une commande

Notre script s'exécute uniquement à partir du répertoire où il se trouve, ou en ajoutant tout son chemin absolu, et avec un « ./ » devant son nom. Nous voulions qu'il s'exécute de la même façon qu'on exécute la commande « ls » c.-à-d. de n'importe quel répertoire et sans « ./ ». Pour cela il faut mettre notre script dans un des répertoires du PATH¹.

Copier en mode root (administrateur) votre script dans un de ces répertoires, tel que /bin, ou /usr/bin, ou /usr/local/bin (ou un autre répertoire du PATH).

Code : Console

```
$ essai.sh  
/home/monpc  
essai.sh
```

8. Exercice:

- Créer une commande appelée maintenant qui permet d'afficher la date et l'heure.
- En utilisant la commande shutdown avec ses options, créer une commande qui permet d'éteindre la machine.
- Écrire un script qui permet d'automatiser les tâches suivantes:

Ce script doit:

- Créer deux répertoires d1 et d2 dans le répertoire en cours.
- En utilisant la commande cat, créer un fichier f1 dont l'utilisateur va saisir son nom et son prénom.
- Copier ce fichier f1 dans les répertoires d1 et d2.

¹Le PATH est une variable système qui indique les répertoires dans lesquelles se trouvent les programmes exécutables. Pour les avoir, il suffit de taper `echo $PATH`.