# Programmation et Scripts Shell

## Appel et exécution de scripts shell

1. Dans votre répertoire personnel, créer le sous-répertoire bin qui sera utilisé pour stocker tous les scripts shell écrits par la suite et se placer dans ce répertoire.
2. Créer le script shell **scriptappel** qui effectue les opérations suivantes :
   * Affecter la chaîne de caractères "abc" à la variable var.
   * Afficher la chaîne de caractères "La variable $var a pour valeur : " suivie du contenu de la variable var.
   * Effectuer une pause de trois secondes.
3. Exécuter le script shell scriptappel de la manière suivante :   
   **bash scriptappel**Après la fin de l'exécution du script, quelle est la valeur de la variable var dans votre environnement shell ? Pourquoi ?
4. Exécuter le script shell **scriptappel** de la manière suivante :  
   **scriptappel**Est-ce possible ? Pourquoi ?
5. Ajouter le droit d'exécution au script scriptappel pour le propriétaire du fichier (Indication : utiliser la commande chmod) et ajouter le chemin de votre répertoire bin à la variable d'environnement PATH s'il n'y est pas déjà.
6. Exécuter le script shell **scriptappel** de la manière suivante :  
   **scriptappel**Après la fin de l'exécution du script, quelle est la valeur de la variable var dans votre environnement shell ? Pourquoi ?
7. Quel shell a été utilisé pour interpréter le script **scriptappel** appelé précédemment ?  
   Faites en sorte que ce soit obligatoirement un shell Bash (Indication : pour imposer un interpréteur plutôt qu'un autre, il suffit de modifier le commentaire spécial en première ligne du script : #! suivi du chemin vers l'interpréteur souhaité).
8. Exécuter le script shell **scriptappel** de la manière suivante :  
   **. scriptappel**Après la fin de l'exécution du script, quelle est la valeur de la variable var dans votre environnement shell ? Pourquoi ?
9. Exécuter le script shell **scriptappel** de la manière suivante :  
   **exec scriptappel**Que se passe-t'il ? Pourquoi ?

## Codes de retour

1. Taper la commande **ls /etc/passwd** et afficher son code de retour. Expliquer la valeur obtenue.
2. Taper la commande **ls flop** et afficher son code de retour. Expliquer la valeur obtenue.
3. Afficher de nouveau le code de retour de la dernière commande. Est-ce le même ?  
   Pourquoi ?
4. Aller dans votre répertoire **bin** et modifier le script **scriptappel** de façon qu'il renvoie un code de retour égal à deux puis le tester (Indication : utiliser **exit**).

## Enchaînement de commandes

1. Afficher la date système et la liste des fichiers présents dans le répertoire courant en une seule ligne de commande.
2. Afficher le contenu du fichier **/etc/hosts** s'il existe (Indication : utiliser le code de retour de la commande ls (en éliminant ses sorties standards **stdout** et erreurs **stderr**) pour savoir si   
   /**etc/hosts** existe puis effectuer un enchaînement de commandes conditionnelles avec les caractères **&&**).
3. Créer le fichier vide **/tmp/flop** s'il n'existe pas (Indication : utiliser le code de retour de la commande ls (en éliminant ses sorties standards **stdout** et erreurs **stderr**) pour savoir si **/tmp/flop** existe puis effectuer un enchaînement de commandes conditionnelles avec les caractères ||).

## Variables spéciales

1. Dans votre répertoire bin, créer le script shell **scriptvars** qui effectue les opérations suivantes :
   * Afficher le nom du script shell.
   * Afficher le PID du script shell.
   * Afficher le PID du processus père (Indication : utiliser la variable **PPID**).

Chaque affichage doit être précédé d'un énoncé (Exemple pour le nom du script : Mon nom est : "Nom du script shell").

Puis modifier les droits du fichier **scriptvars** de façon à lancer son exécution par la simple saisie de son nom.

1. Afficher le PID de votre shell courant puis exécuter le script **scriptvars** des trois manières suivantes :  
   **bash scriptvars  
   scriptvars  
   . Scriptvars**Les résultats sont-ils ceux attendus ?
2. Copier le script **scriptvars** en **scriptargs** et modifier ce dernier de façon à :
   * Afficher le nombre d'arguments passés sur la ligne de commandes
   * Afficher les trois premiers arguments de la ligne de commandes
3. Tester votre script shell **scriptargs** avec les arguments suivants (un seul espace sépare chaque argument) :  
   **a b c d  
   "a b" c d a b c\ d  
   a 'b c' d**
4. Ajouter les opérations suivantes dans le script scriptargs :  
   Décaler les arguments de deux rangs (Indication : utiliser shift).  
   Afficher de nouveau les trois premiers arguments Tester de nouveau le script avec les arguments suivants :   
   **a b c d**

## Tests de chaînes de caractères

1. Dans votre répertoire **bin**, créer le script shell **scriptchaine** qui effectue les tests suivants sur deux chaînes de caractères passées en argument :
   * Si au moins une des deux chaînes de caractères passées en argument est nulle (deux guillemets ""), sortir avec un code de retour égal à un.
   * Afficher si les chaînes de caractères sont identiques ou non.
2. Tester le script shell **scriptchaine** avec les arguments suivants :  
   **abc ""  
   "" abc   
   "" ""  
   abc abc  
   abc ABC  
   abc "abc "**

## Tests et opérations arithmétiques

1. Dans votre répertoire **bin**, créer le script shell **scriptmax** qui retourne le plus grand des deux arguments passés en paramètres.  
   Indication : effectuer les opérations suivantes dans le script :
   * Si le nombre d'arguments est différent de deux, sortir avec un code de retour égal à un.
   * Afficher le nombre le plus grand ou la valeur du premier en cas d'égalité.
2. Tester le script shell **scriptmax** avec les arguments suivants :  
   **12  
   12 34  
   12 6  
   12 12  
   12 " 34"**
3. Dans votre répertoire **bin**, créer le script shell **scriptmin** qui retourne le plus petit des deux arguments passés en paramètres.  
   Indication : effectuer les opérations suivantes dans le script :
   * Si le nombre d'arguments est différent de deux, sortir avec un code de retour égal à un.
   * Afficher le nombre le plus petit ou la valeur du premier en cas d'égalité.
4. Tester le script shell scriptmin avec les arguments suivants :  
   **12  
   12 34  
   12 6  
   12 12  
   12 " 34"**

## Boucle while

1. Écrire le script shell **scriptcompar** qui affiche le plus petit et le plus grand nombre parmi les arguments passés sur la ligne de commandes. Le nombre d'argument doit être supérieur à un (Indication : Réutiliser les scripts **scriptmax** et scriptmin précédemment écrits. Utiliser une boucle **while** avec la commande **shift** pour traiter l'ensemble des arguments).
2. Tester le script shell **scriptcompar** avec les arguments suivants :  
   **1  
   1 2  
   1 2 3  
   3 5 2 6 1 8 7**