

TD séance n° 7

Introduction à la Programmation Shell Unix

4 Exercices

Pour ce premier TD, nous allons nous contenter de faire des petits exercices simples pour apprendre à maîtriser les rudiments de la programmation Shell. Lors de la prochaine séance, nous étudierons les spécificités de la programmation Shell qui en font son intérêt.

Exercice n°1:

Modifiez la variable d'environnement `PATH` pour y ajouter le chemin courant. Attention à ne pas supprimer l'ensemble des chemins de recherche existants.

```
PATH=$PATH:.
```

Exercice n°2:

Comme dans tout nouveau langage de programmation, vous allez faire votre premier script Shell `hello.sh` en faisant le très classique programme « Hello World ! ». L'exécution de votre programme devra afficher ce message à l'écran.

```
#!/bin/bash
echo "Hello World !"
Ne pas oublier de faire le chmod a+x sur le fichier.
```

Refaites le même exercice en mettant le « Hello World ! » dans une variable et en mettant la date courant dans une variable afin que l'exécution de votre script donne le résultat suivant :

```
$ hello2.sh
Hello World ! Nous sommes le lun.  8 déc. 2014 10:17:12
#!/bin/bash
msg="Hello World !"
d=$(date)
echo "$msg Nous sommes le $d"
```

Exercice n°3:

Ecrire un programme Shell qui s'appelle `deuxfois.sh` et qui affiche le message "Entrez un mot : ", lit le mot saisi par l'utilisateur puis affiche ce mot deux fois sur la même ligne.

```
#!/bin/bash
read -p "Entrez un mot : " mot
echo $mot $mot
```

Exercice n°4:

Ecrire un programme Shell qui s'appelle `deuxfois2.sh` et qui affiche le premier paramètre de votre script deux fois sur la même ligne.

```
$ deuxfois2.sh "Mon message"
Mon message Mon message
#!/bin/bash
echo $1 $1
```

Exercice n°5:

Ecrire un programme Shell `cp2fois.sh` prenant trois arguments (3 paramètres) : le premier désigne le nom du fichier dont on veut copier le contenu et les deuxième et troisième arguments sont les noms des fichiers vers lesquels on veut faire la copie du premier. Aucun cas d'erreur ne sera à prendre en compte.

```
$ cp2fois.sh fichier1.txt fichier2.txt fichier3.txt
#!/bin/bash
cp $1 $2
cp $1 $3
```

Exercice n°6:

TD séance n° 7

Introduction à la Programmation Shell Unix

Faites un script `mon_onzieme.sh` qui prend douze paramètres et qui affiche la valeur du onzième.

```
$ mon_onzieme.sh un deux trois quatre cinq six sept huit neuf dix onze douze
#!/bin/bash
echo ${11}
```

Exercice n°7:

Faites un script qui se nomme `copie.sh` et qui affiche le nom du programme, le nombre d'arguments, l'ensemble des arguments qui ont été passés à ce script ainsi que les 2 paramètres sources et destination. Enfin, le programme devra réaliser la copie du fichier source spécifié à la bonne destination.

```
$ copie.sh /etc/passwd X
Nom du programme : ./copie.sh
Nb d'arguments : 2
Tous les arguments : /etc/passwd X
Source : /etc/passwd
Destination : X
#!/bin/bash
echo "Nom du programme : $0"
echo "Nb d'arguments : $# "
echo "Tous les arguments : $*"
echo "Source : $1"
echo "Destination : $2"
cp $1 $2
```

Exercice n°8:

Ecrivez le script `welcome1.sh` qui affiche un message de bienvenue :

```
$ welcome1.sh
Bonjour zorro, bienvenu sur hal9000, nous sommes le 08/12/2014 et il est 10:17
```

Ici, *zorro* est le nom de l'utilisateur qui exécute ce programme, *hal9000* est le nom de la machine sur laquelle on l'exécute. Ces informations sont contenues dans les variables `USER` et `HOSTNAME`.

Les commandes nécessaires sont : `echo`, `date` (rappelez-vous que la lecture de la page de manuel des commandes n'est pas optionnelle ! RTFM !)

```
#!/bin/sh
echo -n "Bonjour $USER, bienvenu sur $HOSTNAME,
echo "nous sommes le `date +%d/%m/%Y` et il est $(date +%H/%M)"
```

Mais où trouver la page de manuelle des commandes internes ?

Dans la page de manuel de votre interprète de commandes bien sûr donc dans notre cas dans le manuel de `bash`.