

Système d'exploitation LINUX

TD/TP 4

Exercice 1 : Que fait ce programme

Que fait le programme shell suivant (remplacez les '...' par le texte adéquat) ?

```
#!/bin/sh
w=`who | grep $1`
if [ -z "$w" ]; then echo "$1 n'est pas ..."; fi
```

Exercice 2 : Compter les utilisateurs

Ecrivez un script Bash nommé **bash_user** qui calcul le nombre d'utilisateur du système qui utilise la **Bash** comme shell par défaut.

L'exécution devra avoir l'affichage suivant à l'écran (si ce nombre d'utilisateur est 13):

```
>Il y a 13 utilisateurs de ce systeme dont le login shell est bash
```

Si il n'existe aucun utilisateur qui utilise le bash, le script affichera le message :

```
>Il n'y a aucun utilisateur dont le login shell est bash.
```

Exercice 3: Changement d'extension

Écrivez le script "change_ext" permettant de renommer un ensemble de fichiers.

Par exemple :

```
change_ext '.c' '.bak'
```

aura pour effet de renommer tous les fichiers d'extension .c en .bak. les fichiers f1.c et f2.c deviennent f1.bak et f2.bak.

NB : Utilisez la commande **basename**.

basename est une commande Unix qui renvoie le préfixe du fichier placé en paramètre, selon le suffixe mentionné ensuite.

Exemple :

```
smi@ubuntu:~$ ls -l ex.sh
-rwxrwxr-x 1 smi smi 335 Dec  6 20:07 ex.sh
smi@ubuntu:~$ basename ex.sh .sh
ex
```

Exercice 4 : Affichage de date

La commande **date** permet d'afficher la date actuelle du système :

```
smi@ubuntu:~$ date
Tue Dec  8 06:41:58 PST 2015
```

Vous remarquez que l'affichage de la date contient des abréviations en anglais. On désire créer un script **horloge** qui permet d'embellir l'affichage de ces informations de tel sorte à avoir un affichage de la date sous cette forme :

```
smi@ubuntu:~$ ./horloge.sh
on est le Mardi 8 Décembre 2015
l'horloge indique 06 heures, 41 minutes et 58 secondes
```

Pour cela, récupérer le résultat de la commande **date** en interne à travers les variables spécial (en utilisant la commande **set**).

Puis, à l'aide de la structure de contrôle **case**, remplacez les abréviations des jours et des mois par les mots entiers en français.

Ensuite afficher la date sous la forme désirée

Exercice 5: Renommage interactif de fichiers d'un répertoire

Le but de cet exercice est de proposer un script qui permet de renommer interactivement les fichiers d'un répertoire passer en argument. Pour cela :

1. Utiliser la boucle `for` pour afficher, un par un tous les fichiers du répertoire courant.
2. Poser la question "Nouveau nom (ou) ?" et lire la réponse (`echo` et `read`).
3. Renommer le fichier selon la réponse ou traiter le fichier suivant.
4. Compléter le script pour opérer dans le répertoire de travail de l'utilisateur (script sans argument) ou dans un répertoire quelconque transmis comme unique argument.

Le script vérifie la validité des arguments en effectuant les tests suivants :

- test de la syntaxe d'appel (message si le nombre d'arguments d'appel est incorrect),
- test de l'existence et de la lisibilité des objets transmis en arguments,
- test de la préexistence d'une entrée cible (fichier ou répertoire).

Ainsi, la commande `mv f1 f2` peut écraser l'élément `f2` si celui-ci existe. On peut donc refuser et traiter l'élément suivant ou demander un autre nom et confirmation avant d'écraser l'élément.