

CORRECTION ATELIER

RECAP : Les bases de la programmation shell

Ecrire une procédure script shell

1. Créez un script shell nommé **parametres** qui affiche les cinq lignes suivantes, en utilisant les variables prédéfinies du shell pour remplacer les "_____" par les paramètres transmis. Exécutez le script avec les arguments 10 100 1000.

Le nom de la procédure est _____.
Le premier paramètre reçu est _____.
Le deuxième paramètre est _____.
Le troisième paramètre est _____.
En tout il y a _____ paramètres transmis.

```
$ vi parameters
echo Le nom de la procédure est $0.
echo Le premier paramètre reçu est $1.
echo Le deuxième paramètre est $2.
echo Le troisième paramètre est $3.
echo En tout il y a $# paramètres transmis.
$ chmod +x parameters
$ . parameters 10 100 1000
```

2. Créez une procédure **veriffic**, qui affiche le contenu du fichier **parametres** à condition qu'il **existe**. Utilisez un opérateur d'exécution conditionnelle.

```
$ vi veriffic
ls parameters && cat parameters
$ chmod +x veriffic
$ . veriffic
```

3. Modifiez la procédure **veriffic** pour remplacer le nom du fichier **parametres** par un nom de fichier qui n'existe pas **nonfic** (soyez certains qu'il n'y a pas de fichiers nommés **nonfic** dans votre répertoire courant). Toujours en utilisant un opérateur d'exécution conditionnelle, si la commande **ls** échoue, alors affichez un texte d'erreur **Le fichier n'a pas été trouvé**. Lancez la procédure.

```
$ vi veriffic
ls nonfic && cat nonfic || echo "Le fichier n'a pas été trouvé"
$ . veriffic
```

4. Modifiez la procédure **veriffic** pour que le message d'erreur affiché par la commande **ls** n'apparaisse pas à l'écran. Lancez la procédure.

```
$ vi veriffic
```

```
ls nonfic 2> /dev/null && cat nonfic || echo "Le fichier n'a pas été trouvé"
$ . veriffic
```

5. Modifiez la procédure **veriffic** pour pouvoir lui transmettre un argument sur la ligne de commande. Cet argument est utilisé, dans le code de la procédure, comme paramètre des commandes **ls** et **cat**. Lancez deux fois la procédure script shell, une fois avec parametres, la seconde fois avec **nonfic**.

```
$ vi veriffic
ls $1 2> /dev/null && cat $1 || echo "Le fichier n'a pas été trouvé"
$ . veriffic parametres
$ . veriffic nonfic
```

Utilisation de for, test et if

6. En utilisant la boucle for, modifiez la procédure **veriffic** pour qu'elle puisse recevoir plusieurs noms de fichier en paramètre au lieu d'un seul. Si les fichiers existent alors affichez leur contenu. Si les fichiers n'existent pas, alors affichez un message d'erreur comme quoi le fichier n'a pas été trouvé. Repérez quelques noms de fichier présents dans votre répertoire courant. Lancez la procédure en indiquant des noms de fichiers existants et de fichiers inexistantes.

```
$ vi veriffic
for x in $*
do
    ls $x 2> /dev/null && cat $x || echo $x pas trouvé
done
$ ls
$ veriffic fic1 fic2 fic3
(où fic1 fic2 fic3 sont remplacés par des noms de fichier, existants ou pas, dans votre répertoire courant).
```

7. Modifiez votre procédure **veriffic** pour utiliser l'opérateur **if** et la commande **test**, à la place des opérateurs d'exécution conditionnelle, pour vérifier que le nom de fichier est celui d'un fichier existant ou pas. Lancez la procédure comme vous l'avez fait à l'étape précédente.

Astuce : les codes retours sont utilisés dans cette procédure.

```
$ vi veriffic
for x in $*
do
    ls $x 2> /dev/null
    if [[ $? -eq 0 ]]
    then cat $x
    else echo $x pas trouvé
    fi
done
$ veriffic fic1 fic2 fic3
```

Utilisation de while et expr

8. Créez une procédure nommée **manger**, qui utilise une boucle **while** infinie, pour afficher **A table !** toutes les deux secondes. Lancez ce script shell. Quand vous en avez assez, interrompez la boucle.

```
vi manger
while true
do
    echo A table !
    sleep 2
done
$ chmod +x manger
$ . manger
<Ctrl-c>
```

9. Directement sur la ligne de commande, affichez le résultat de la multiplication de 5 fois 6.

```
$ expr 5 \* 6
```

10. Maintenant, toujours en utilisant **expr**, créez une procédure nommée **math** qui accepte deux nombres comme arguments sur la ligne de commande et qui affiche leur multiplication. Lancez la procédure pour calculer 5 fois 6. Testez avec d'autres nombres.

```
$ vi math
expr $1 \* $2
$ chmod +x math
$ . math 5 6
```

Application

Pour chaque extension **.conf**, **.cfg** et **.d**, indiquez s'il y a plus de **10** fichiers ou non, dont le nom se termine avec cette extension dans le répertoire **/etc**.

Réponse

La solution utilise une boucle **for** pour parcourir avec la variable **\$ext** les différentes extensions proposées dans l'énoncé.

```
$ vi verifext
for ext in .conf .cfg .d
do
    [[ $(ls -d /etc/*$ext | wc -l) -gt 10 ]] && echo "il y a plus de
10 fichiers se terminant par $ext" || echo "il y a moins de 10
fichiers se terminant par $ext"
done
$ chmod +x verifext
$ . verifext
```

La commande **ls -d /etc/*\$ext | wc -l**, qui retourne le nombre de fichiers correspondants, est substituée (**\$(...)**) par son résultat dans le test (**[[... -gt 10]]**) qui, lui, détermine s'il y en a plus de 10.

Si ce test est vrai (**&&**), on exécute la commande **echo "...plus..."** ; sinon (**| |**), on exécute la suivante : **echo "...moins..."**.