Correction du TP5 : Programmation Shell (2)

Préparation: Placez-vous sous votre répertoire de connexion et créez un nouveau répertoire nommé TP5 dans lequel vous travaillerez pour répondre aux exercices suivants.

Exercice 1 :

1. Écrire un script Shell "**Inverse**" qui permet d'afficher l'inverse d'un mot donné en argument.
2. Utiliser ce dernier script pour élaborer un nouveau script "**Pal**" permettant de vérifier si le mot passé en argument est palindrome ou non.

```
1. #!/bin/sh
   if [ $# -eq 1 ]
   then

        lg=`expr length $1`
         while [ $lg -gt 0]
         do
                c=`expr substr $1 $lg 1`
                echo -n c
                lg=`expr $lg - 1`
        done
   else
        echo "erreur de parameters"
   fi

2. Appel d'un script par un autre script
#!/bin/sh
   if [ $# -eq 1 ]
   then
   inv=sh inverse $1
        if [ $inv = $1 ]       # comparaison de deux chaînes
        then
                echo palindrome
        else
                echo non palindrome
        fi
   else
        echo "erreur de parametres"
   fi
```

Exercice 2 :

En utilisant la commande **find** et la boucle **while,** écrire un script Shell "CherchFich" qui permet qui permet de réaliser un ensemble d'opérations sur un répertoire passé en argument (par défaut le répertoire courant). Ces opérations sont les suivantes:

1. L'affichage du menu suivant :

********** Menu **********

<1> Affichage de la liste des fichiers spéciaux
<2> Affichage des répertoires de taille supérieure à 512 Octets
<3> Affichage des fichiers qui ont été accédé depuis 8 jours
<4> Affichage des fichiers qui ont été créés depuis 3 jours
<5> Affichage des fichiers qui n'ont pas été modifiés depuis 5 jours
<6> Affichage des contenus des fichiers ordinaires qui ont les droits d'accès 755

<7> Quitter
****

2. La demande à l'utilisateur de choisir une option de ce menu.
3. L'exécution la commande correspondante au choix.

N.B : le script doit répéter ces opérations tant que l'utilisateur n'a pas demandé de quitter.

```
#!/bin/sh
if [ $# -gt 1 ]
then
        echo Trop de paramètres
else
 if [ $# -eq 0 ]
 then

      rep=.

 else

      rep=$1

 fi

 choix=0

 while [ $choix -ne 7 ]

 do

   echo "********************** Menu ****************************"

   echo "<1> Affichage de la liste des fichiers spéciaux"

   echo "<2> Affichage des répertoires de taille supérieure à 512
Octets "

   echo "<3> Affichage des fichiers qui ont été accédé depuis 8
jours "

   echo "<4> Affichage des fichiers qui ont été crées depuis 3
jours"

   echo "<5> Affichage des fich qui n'ont pas été modifiés depuis 5
jours "

   echo "<6> Affichage des contenus des fich ordinaires dont le
droit 755"

   echo "<7> Quitter "

   echo "*****************************************************"

   read choix

   case $choix in

   1)find  $rep \( -type c -o -type b -o -type s -o -type s \) -print;;

   2)find  $rep -type d -size +512c -print;;
  #find  $rep \(-type d -a -size +512c\) -print;;
```

```
   3)find  $rep -atime 8 -print;;

   4)find  $rep -ctime 3 -print;;

   5)find  $rep -not -mtime 5 -print;; #
   find  $rep !-mtime -5 -print;;

   6)find  $rep -type f -perm -755 -exec cat{} \;;;

   7) exit ;;

   *)echo "choix incorrect" ;;
   esac
 done
fi
```

Exercice 3 :

En utilisant la structure **select**, écrire un script "**Formattage**" qui permet de réaliser le menu suivant :

********** Menu **********

<1> Afficher un fichier donné à l'envers
<2> Numéroter les lignes d'un fichier donné
<3> Afficher le contenu d'un fichier donné sous une autre forme indiquée par l'utilisateur
<4> Réduire les espaces entre les mots d'un fichier donné à un espace
<5> Convertir les tabulations d'un fichier donné en espaces
<6> Fusionner les lignes de deux fichiers donnés
<7> Découper un fichier donné selon un nombre de lignes indiqué par l'utilisateur
<8> Quitter

****

```
#!/bin/sh
PS3="euillez choisir un numéro"
Select choix in \
"Afficher un fichier donné à l'envers"\
"Numéroter les lignes d'un fichier donné"\
"Afficher le contenu d'un fichier donné sous une autre forme indiquée par
l'utilisateur"\
"Réduire les espaces entre les mots d'un fichier donné à un espace"\
"Convertir les tabulations d'un fichier donné en espaces"\
"Fusionner les lignes de deux fichiers donnés"\
"Découper un fichier donné selon un nombre de lignes indiqué par
l'utilisateur"\
"Quitter";
do
case $REPLY in
1)echo "Donner un fichier"
  read fich
  tac $fich ;;
2)echo "Donner un fichier"
  read fich
  nl $fich ;;
3)echo "Donner un fichier"
  read fich
  echo "Donner le format d'affichage : o : pour valeur octale, x : pour
valeur hexadécimale, c : pour caractères ASCII"
```

```
  read forma
  od –t $forma $fich ;;
4)echo "Donner un fichier"
  read fich
  fmt –u $fich ;;
5)echo "Donner un fichier"
  read fich
  expand –t $fich ;;
6)echo "Donner le premier fichier"
  read fich1
  echo "Donner le deuxième fichier"
  read fich2
  paste $fich1 $fich2 ;;
7)echo "Donner un fichier"
  read fich1
  echo "Donner le nombre de lignes"
  read nbr
  split -$nbr $fich1 f_;;

8) exit ;;
*) echo "choix incorrect" ;;
esac
done
```

Complément du TP5

find rép_de_départ [critères] [option de cdes]

Cette commande sert à rechercher des fichiers. Tous les répertoires spécifiés (paramètre rép_de_départ) et leurs sous-répertoires sont parcourus de manière récursive.

Critère :

-name fichier : recherche par nom de fichier (caractères spéciaux : * , ? , […])

-type type : recherche par type de fichier (f : fichier ordinaire, d : répertoire, c : fichier périphérique de type caractère, b : fichier périphérique de type bloc, s : socket).

-user nom : recherche par propriétaire.

-group nom : recherche par groupe.

-size nombre : recherche par taille (pour les octets, il faut ajouter un c).

-empty peut être utilisé en remplacement de -size 0.

-atime jour : recherche par date de dernier accès.

-mtime jour : recherche par date de dernière modification.

-ctime jour : recherche par date de création.

-perm droits : recherche par droits d'accès.

-links nombre : recherche par nombre de liens.

-inum permet une recherche par numéro d'inode. Elle est utile dans le cas d'une recherche de tous les liens portant un même numéro d'inode.

Options de commande :

-print : affiche le résultat.

-exec : introduit une commande si un fichier est trouvé.

-ok : même chose que exec mais avec une demande de confirmation.

Remarque :

- Pour toutes les options en dehors de –name, -type, -user, -group et –perm, les noms d'options sont complétés par des valeurs. Ces nombres peuvent aussi être précédés des signes + (supérieur à) ou – (inférieur à).
- La commande placée derrière –exec doit se terminer par ;. Comme les points virgules sont des caractères spéciaux, ils doivent être masqués par un backslash.
- Si, dans la commande placée derrière –exec, vous voulez accéder au fichier qui vient d'être trouvé, vous utiliserez l'abréviation {} (deux accolades l'une après l'autre).
- Si plusieurs options sont spécifiées, elles sont liées par –a ou –o. On utilise aussi ! pour la négation. Ces options liées doivent être placées entre parenthèses. Comme ces parenthèses sont des caractères spéciaux, il faut les masquer par un backslash.

tac: affiche le contenu inversé d'un fichier.

exemples : tac f1

nl : numérote les lignes d'un fichier.

exemples : nl f1

ls | nl –s')'

od : afficher le contenu d'un fichier en octal ou sous d'autres formes (decimal, hexadecimal, etc.)

option : -t type

Le type peut être :

a : caractères littéraux
c : caractères ASCII
o : valeurs octales
u : valeurs décimales non signées
x : valeurs hexadécimales
exemples : od –t o file1
od –t x file1
fmt : formater les paragraphes dans un fichier.
options :
-u : espacement uniforme. Réduire les espacements entre les mots à un espace.
-w : remplir les lignes jusqu'à la largeur mentionnée (par défaut 75 caractères)
exemple :
fmt –w 80 myfile.txt > myfile80wide.txt
paste : regrouper les lignes de différents fichiers.
exemples :
file1
1
2
3
file2
A
B
C
paste file1 file2
1 A
2 B
3 C
paste –d'@' file1 file2
1@A
2@B
3@C
paste –s file1 file2
1 2 3
A B C
split : découper un fichier en différentes partie.
exemples :
file1
1 one
2 two
3 three
4 four
5 five
6 six
split –2 file1 splitout_ : créer trois fichiers splitout_aa, splitout_ab, splitout_ac.

split –b 1.4m grosfichier petitfichier_ : découper un fichier en plusieurs de taille maximale d'une disquette.

expand : convertir les tabulations d'un fichier en espaces.

unexpand : fait le processus inverse de expand.