

## Correction TD4 : Les scripts shell

### Exercice 1 :

1. Afficher toutes les variables d'environnement du système. Et compter les.
2. Afficher le contenu de la variable d'environnement « PATH ».
3. Afficher le nom de l'utilisateur courant suivi de son répertoire personnel.
4. Définir une variable locale: annee=2021.
5. Ouvrir une nouvelle session shell (taper la commande sh)
6. Taper la commande echo \$annee. Que constatez-vous ?
7. Exporter la variable annee et refaire 6.
8. Rajouter cette variable au fichier .bashrc et refaire 6.

### Exercice 2 :

Ecrire un script shell qui permet de changer l'extension des fichiers .cpp qui se trouvent dans un répertoire passé en argument par une deuxième extension passé en argument aussi.

```
#!/bin/bash
if [ $# != 2 ]
then
    echo "syntaxe: $0 repertoire extension"
else
    if test -d $1
    then
        for x in $1/*.cpp ; do
            #echo $x
            name=`echo "$x"|cut -d "." -f1`
            echo "$name"
            mv "$x" "$name.$2"
        done
    else
        echo "$1 doit etre un repertoire"
    fi
fi
```

### Solution 2 :

```
#!/bin/bash
#on teste si nb parametres = 2
#les deux entrees sont $1= ./DOSSIER $2= extension
if test $# -eq 2
then
    #on teste le 1er param est un dossier
    if test -d $1
    then
        #on parcour le contenu du dossier et on itere sur les noms de fichiers qui se
        terminent avec un .cpp
        for x in $1/*.cpp ; do
            # on recupere la 3eme colonne de x qui va etre de la forme
```

```
(./DOSSIER/FICHIER.cpp) puis la 1ere colonne avec le delim
# '.' et donc la variable a possedera (FICHIER)
a=`echo $x | cut -d '/' -f 3 | cut -d '.' -f 1`
#on move les fichiers x vers le dossier $1 qui sera sous forme (./DOSSIER)
mv "$x" "$1/$a.$2"
done
else
echo "Error: only directories are accepted"
fi
else
echo "Error: you should specify two parameters"
fi
```

### Exercice 3 :

1. Créer la commande **crefic1** obéissant à la syntaxe suivante :  
# **crefic1** nomfichier  
permettant de créer un fichiers nomfichier dans le répertoire courant.

```
#!/bin/sh
if [ $# -ne 1 ]
then
    echo "syntaxe: $0 file_name "
    exit 1
else
    touch $1
fi
```

2. Créer la commande **crefic2** obéissant à la syntaxe suivante :  
# **crefic2** nomfichier  
permettant de créer un fichiers nomfichier dans le répertoire courant après validation de l'utilisateur:  
\$./crefic2 toto  
voulez vous créer toto ?  
Oui  
toto est crée  
\$./crefic2 test  
voulez vous créer test?  
Non

```
#!/bin/sh
if [ $# -ne 1 ]
then
    echo "syntaxe: $0 file_name "
    exit 1
else
    echo "voulez vous créer $1 (y/n)"
    read ans

    while [ "$ans" != "y" ] && [ "$ans" != "yes" ] && [ "$ans" != "n" ] && [ "$ans" != "no" ]
    do
        echo "repondre par y/yes ou n/no :"
        read ans
    done

    if [ "$ans" = "y" ] || [ "$ans" = "yes" ]
    then
        if test -f $1
        then
            echo "Choisir un autre nom de fichier: celui la existe déjà! "
        else
            touch $1
            echo "$1 est crée"
        fi
    fi
fi
```

3. Créer la commande **crefic3** obéissant à la syntaxe suivante :

# **crefic3** nom N

*Son rôle est de créer un ensemble de fichiers nom1, nom2, ...nomN. La création de chaque fichier doit être validée en interactif par l'utilisateur.*

```
#!/bin/sh
if [ $# -ne 2 ]
then
    echo "syntaxe: $0 <file_name> <Number> "
    exit 1
elif [ $2 -eq 0 ];
then
    echo "le 2eme argument doit être different de 0"
    exit 2
else
    nb=$2
    while [ $nb -gt 0 ]
    do
        echo "voulez vous créer le fichier $1$nb (y/n)? "
        read ans
        while [ "$ans" != "y" ] && [ "$ans" != "yes" ] && [ "$ans" != "n" ] && [ "$ans" != "no" ]
        do
            echo "repondre par y/yes ou n/no :"
            read ans
        done

        if [ "$ans" = "y" ] || [ "$ans" = "yes" ]
        then
            if test -f $1$nb
            then
                echo "Choisir un autre nom de fichier: celui la existe déjà! "
            else
                touch $1$nb
            fi
        fi

        nb=`expr $nb - 1 `
    done
fi
```

## Exercice 4 :

Ecrire un script nommé **indice** qui affiche l'indice de son premier paramètre dans la liste des paramètres qui suivent.

## Exemples :

```
$ indice toto tata titi toto tutu
L'indice de toto dans la liste est : 3
$ indice 9 2 8 6 9 5 0
L'indice de 9 dans la liste est : 4
```

```
#!/bin/bash
```

```
if [ $# -ge 2 ];then
```

```
    ARG=$1 #sauvegarde la valeur du 1er argument
```

```
    shift # $1 désigne le 2eme argument
```

```
    ind=0
```

```
    for argi in $@
```

```
    do
```

```
        ind=`expr $ind + 1`
```

```
        if [ "$ARG" = "$argi" ]
```

```
        then
```

```
            echo "l'indice de $ARG est: $ind"
```

```
        fi
```

```
    done
```

```
else
```

```
    echo "syntaxe:$0 arg1 arg2"
```

```
fi
```

```
_
```

## Exercice 5 :

Ecrire un script qui affiche le menu suivant, et ensuite demande à l'utilisateur de rentrer un nombre (de 1 à 4), et qui affiche le résultat de l'exécution de la commande entre les parenthèses selon l'entrée de l'utilisateur :

1. Lister le contenu du répertoire (\$ls -l)
2. Lister les processus en cours d'exécution (\$ps aux)
3. La date d'aujourd'hui (\$date)
4. Quitter

```
#!/bin/bash
```

```
# Menu
```

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

```
echo "1- Lister le contenu du repertoire courant"
```

```
echo "2- Lister les processus en cours d'exécution"
```

```
echo "3- La date d'aujourd'hui"
```

```
echo "4- Quitter"
```

```
echo "***** "
```

```
echo "Donner votre choix: "
```

```
read choix
```

```
case $choix in
```

```
    1) if [ $# -ne 1 ]
```

```
    then
```

```
        echo "error: syntaxe $0 repertoire"
```

```
    else
```

```
        echo "Le contenu du repertoire courant est: "
```

```
        echo $(ls -l .) #`ls -l .`
```

```
    fi;;
```

```
    2)echo " Lister les processus en cours d'exécution: "
```

```
        ps aux ;;
```

```
    3) echo `date`;;
```

```
    4) exit;;
```

```
esac
```

## Exercice 6 :

Ecrire un script qui prend comme argument un fichier et transforme toutes les lettres minuscules en majuscules.

```
#!/bin/bash

if [ $# -ne 1 ]
then
    echo "le nombre d'arguments doit être égal a un "
    echo "syntaxe: $0 file "
else
    # cat $1|tr '[:lower:]' '[:upper:]'|tee "$1Maj"
    sed -i 's/./\U&/' $1
fi
```

## Exercice 7 :

Ecrire un script "test-fichier.sh", qui précisera le type du fichier passé en argument, ses permissions d'accès pour l'utilisateur. Si le script ne reçoit aucun paramètre, il devra afficher un message d'erreur.

*Exemple de résultats :*

Le fichier /etc est un répertoire

"/etc" est accessible par root en lecture écriture exécution

Le fichier /etc/smb.conf est un fichier ordinaire qui n'est pas vide

"/etc/smb.conf" est accessible par toto en lecture.

```
#!/bin/bash
if [ $# -ne 1 ]; then
    echo "Usage : $0 <nom_fichier>"
    exit 1
fi
if [ ! -e "$1" ]; then
    echo "Le fichier '$1' n'existe pas"
    exit
fi

if [ -f "$1" ]; then
    echo "$1 est un fichier ordinaire qui n'est pas vide"
elif [ -d "$1" ]; then
    echo "$1 est un répertoire"
else
    echo "$1 n'est ni un fichier ni un répertoire"
fi

if [ -r "$1" ]; then
    if [ -w "$1" ]; then
        if [ -x "$1" ]; then
            echo "$1 est accessible en lecture ecriture exécution"
        else
            echo "$1 est accessible en lecture ecriture"
        fi
    elif [ -x "$1" ]; then
        echo "$1 est accessible en lecture exécution"
    else
        echo "$1 est accessible en lecture"
    fi
elif [ -w "$1" ]; then
    if [ -x "$1" ]; then
        echo "$1 est accessible en ecriture exécution"
    else
        echo "$1 est accessible en ecriture"
    fi
elif [ -x "$1" ]; then
    echo "$1 est accessible en exécution"
else
    echo "$1 est inaccessible"
fi
```

## Exercice 8 :

Ecrire un script qui prend en argument le nom de plusieurs fichiers et qui affiche le menu ci-dessous :

Voulez-vous créer une archive

- 1) avec l'extension .tar.gz
- 2) avec l'extension .tar.bz2

Ce script doit créer une archive contenant les fichiers en arguments, avec une extension qui dépend de la réponse de l'utilisateur.

Lister le contenu d'une archive : **tar tvjf ex8-archive.tar.bz2**

**tar tvf ex8.tar.gz**

```
#!/bin/bash
# Menu
echo "*****Menu ***** "
echo "Voulez-vous créer une archive"
echo "1) avec l'extension .tar.gz"
echo "2) avec l'extension .tar.bz2"
echo "3) Quitter"
echo "***** "
if [ $# -ne 0 ]
then

echo "Donner votre choix: "
read choix

if [ $choix -eq 1 ]
then
    tar cvfz ex8.tar.gz $*

elif [ $choix -eq 2 ]
then

    tar cvjf ex8-archive.tar.bz2 $*

else

    exit

fi

else
    echo "syntaxe: $0 <file1> <file2>....<filen>"
fi
```