

TP2- Langage de programmation Shell

Objectif:

L'objectif visé de ce TP est:

- Clarifier et expliquer aux intervenants les notions de base d'un Shell Script.

Travail demandé:

A. Manipulation des variables

1. Ecrire un programme **AfficheFormulaire** qui demande de saisir votre nom, prénom et âge ensuite permet de les afficher.

```
1 #!/bin/bash
2 read -p 'Entrez votre nom : ' nom
3 read -p 'Entrez votre prenom : ' prenom
4 read -p 'Entrez votre age : ' age
5 echo " $nom $prenom $age "
```

```
esprit@esprit:~/TP2$ gedit Ex1
esprit@esprit:~/TP2$ ls -l Ex1
-rw-r--r-- 1 esprit esprit 162 févr.  3 14:55 Ex1
esprit@esprit:~/TP2$ chmod u+x Ex1
esprit@esprit:~/TP2$ ./Ex1
Entrez votre nom : mohamed
Entrez votre prenom : ben ali
Entrez votre age : 19
mohamed ben ali 19
```

2. Ecrire un programme **copiedossier** qui prend en argument deux noms de dossiers A et B et qui permet de les créer et de copier l'un dans l'autre.

```
1 #!/bin/bash
2 mkdir $1 $2
3 cp -r $1 $2
4 echo "copied "
```

```
esprit@esprit:~/TP2$ ls
arithmetique  chaine2  chaine4  chaine6  copierdossier  EX3  tableau
chaine1       chaine3  chaine5  chaine8  Ex1           shell
esprit@esprit:~/TP2$ ./copierdossier rep1 rep2
copied
esprit@esprit:~/TP2$ ls
arithmetique  chaine3  chaine6  Ex1  rep2
chaine1       chaine4  chaine8  EX3  shell
chaine2       chaine5  copierdossier  rep1  tableau

esprit@esprit:~/TP2$ cd rep2
esprit@esprit:~/TP2/rep2$ ls
rep1
```

3. Ecrire un programme **effacerFichier** qui détruit le fichier ou dossier passés en paramètre, mais, avant de l'effacer, le programme demande une confirmation pour effacement.

```
1#!/bin/bash
2      rm -ri $1

esprit@esprit:~/TP2$ ls
arithmetique  chaine3  chaine6  effacerFichier  rep1  tableau
chaine1       chaine4  chaine8  Ex1            rep2
chaine2       chaine5  copierdossier  file1          shell
esprit@esprit:~/TP2$ ./effacerFichier file1
rm : supprimer 'file1' du type fichier vide ? o
esprit@esprit:~/TP2$ ls
arithmetique  chaine3  chaine6  effacerFichier  rep2
chaine1       chaine4  chaine8  Ex1            shell
chaine2       chaine5  copierdossier  rep1            tableau
```

B. Tableaux

1. Ecrire un programme **Etudiant** qui permet de créer un tableau contient cette liste des noms « ahmed, ali , mariem, soulaima, salim » et d'afficher le premier et le deuxième nom

```
1#!/bin/bash
2tableau=( ahmed ali mariem soulaima salim )
3echo ${tableau[0]}
4echo ${tableau[1]}

esprit@esprit:~/TP2$ ./Etudiant
ahmed
ali
esprit@esprit:~/TP2$
```

- Ajouter les noms « tarek, samer » et afficher le nombre des étudiants.

```
tableau[${#tableau[*]}]=tarek
tableau[${#tableau[*]}]=samer
echo ${#tableau[*]} #nombre des étudiants
esprit@esprit:~/TP2$ ./Etudiant
7
```

- Ajouter le nom du votre délégué au début du tableau.
- Afficher Tous les éléments du tableau.

```
echo ${tableau[*]} #affichage
tableau=( délégué ${tableau[*]} )
echo ${tableau[*]} #affichage
esprit@esprit:~/TP2$ ./Etudiant
ahmed ali mariem soulaima salim
délégué ahmed ali mariem soulaima salim
```

C. Expr : arithmétique

- Ecrire un programme **arithmétique** qui prend en paramètre 2 entiers et afficher les résultats des opérations arithmétiques suivantes :

addition	multiplication
soustraction	division

```
1#!/bin/bash
2addition=`expr $1 + $2`
3echo $addition
4soustraction=`expr $1 - $2`
5echo $soustraction
6multiplication=`expr $1 \* $2`
7echo $multiplication
8division=`expr $1 / $2`
9echo $division

esprit@esprit:~/TP2$ ./arithmetique 12 6
18
6
72
2
```

D. Expr : manipulation des chaînes

1. Ecrire un programme **chaines** qui prend en paramètre une chaîne de caractères et afficher sa longueur

```
#!/bin/bash
chaîne=BonjourEsprit2023-2024
echo ${#chaîne}
echo `expr length $chaîne`
echo `expr "$chaîne" : '.*'`

esprit@esprit:~/TP2$ ./chaîne1
22
22
22
```

2. On suppose que la chaîne de caractères est **helloWorld**, écrire un programme **souschaînes** qui permet d'afficher la Longueur de 1er sous-chaînes

```
#!/bin/bash
chaîne=helloWorld
#      |----|
#      12345

echo `expr match "$chaîne" '.....'`
echo `expr "$chaîne" : '[a-z][a-z]ll.'`

esprit@esprit:~/TP2$ ./chaîne2
5
5
```

3. Ecrire un programme **Affichage-sous-chaînes** qui prend en paramètre la chaîne **helloWorld** et permet d'afficher la deuxième sous chaîne.

```
#!/bin/bash
chaîne=helloWorld
echo ${chaîne:5:5}

esprit@esprit:~/TP2$ ./chaîne3
World
```

4. Ecrire un programme **Position-numérique** qui permet de donner la position du deuxième sous chaîne de la chaîne **HelloWorld**.

```
#!/bin/bash
chaîne=helloWorld
#      12345678...

echo `expr index "$chaîne" r`
```

```
esprit@esprit:~/TP2$ ./chaine4
8
```

5. Ecrire un programme **Last_Extraction** qui permet d'extraire les 5 derniers caractères de la chaine **helloWorld**.

```
#!/bin/bash
chaine=helloWorld
echo `expr "$chaine" : '.*\(. . . .\)`

esprit@esprit:~/TP2$ ./chaine5
World
```

6. Ecrire un programme **Suppression** qui permet de Supprimer la plus petite correspondance entre h et o à partir du début de la chaine helloworld.
7. Ecrire un programme **Suppression** qui permet de Supprimer la plus grande correspondance entre h et o à partir du début de la chaine helloworld.

```
1#!/bin/bash
2chaine=helloworld
3#      |----|
4#      |-----|
5
6echo ${chaine#h*o}
7# Supprime la plus petite correspondance entre 'h' et 'o'.
8
9echo ${chaine##h*o}      # abc
10# Supprime la plus grande correspondance entre 'h' et 'o'.
|esprit@esprit:~/TP2$ ./chaine6
|world
|rld
```

8. Ecrire un programme **Remplacement** qui permet de remplacer la première correspondance de la sous chaine **world** de la chaine **helloworld** avec **esprit**.

```
#!/bin/bash
chaine=helloWorld

echo ${chaine/World/esprit}

|esprit@esprit:~/TP2$ ./chaine8
|helloesprit
```