

## 1-Exercice

Soit le fichier **agenda.txt** suivant :

Anne-12, rue La Rochelle-17150-0589787878

Sofia -10,Bd La sagesse -67150-0789676767

Paul -07,rue Rome -75150-0789757575

Nour -02,impasse Lumière-10150 -0789101010

1.1 Donne la commande **sed** qui remplace dans les lignes 2 et 3 tous les caractères '-' par ':'

1.1 `sed 2, 3"s/-/ : /g" agenda.txt`

1.2 Donner la commande **awk** qui affiche le contenu du fichier **agenda.txt** sous la forme :

0589787878 \*\* Anne \*\*12, rue La Rochelle \*\* 17150

0789676767 \*\* Sofia \*\*10, Bd. La sagesse \*\* 67150

0789757575 \*\* Paul \*\*07,rue Rome \*\* 75150

0789101010 \*\* Nour \*\*02,impasse Lumière \*\*10150

1.2 `awk -F "-" ' { print $4 "***" $1 "***" $2 "***" $3} ' agenda.txt`

## 2-Exercice

Ecrire un script en bash qui accepte comme argument le nom de fichier **agenda.txt** (voir exercice1) et qui exécute les actions suivantes :

-récupère les noms des futurs utilisateurs (1<sup>re</sup> colonne du fichier)

-teste pour chaque nom récupéré, s'il existe un répertoire avec ce nom dans **/home**

-si oui afficher le message « l'utilisateur existe » puis afficher les processus qui lui appartient.

-Sinon créer un répertoire portant ce nom dans **/home**

-ajouter une ligne dans le fichier **/etc/passwd** sous la forme :

nomUtilisateur:x:id:500 :/home/nomUtilisateur:/bin/bash

**id** doit auto-incrémente automatiquement en partant de 1000

```
#!/bin/bash
utilisateurs=$(cut -d '-' -f1 $1)
echo $utilisateurs
id=100
for u in $utilisateurs
do
    if test -d /home/$u
    then
        echo $u existe déjà
        ps -u $u
    else
        mkdir /home/$u
        echo $u:x: ( ($id+1) ): 500 :: /home/$u : /bin/bash >> /etc/passwd
    fi
done
```

### 3-Exercice

Sur un serveur Linux, donner la commande (en local) qui crée le dump d'une base de données **postgres** de nom **BD\_RT2** ( on considère que l'administrateur postgres a pour mot de passe **secret**), et il a positionné le dump, nommé du nom de la base avec l'extension **.dump**, dans un répertoire **sauvegardeBD** situé dans **/home**.

```
pg_dump -U postgres -psecret -h localhost BD_RT2 > /home/sauvegardeBD/BD_RT.dump
```

### 4-Exercice

Ecrire la commande qui fait un archivage compressé (commande **tar**) du répertoire **sauvegardeBD** vers un lecteur de bandes **rft0**.

```
4. tar -cvf /dev/rft0 /home/sauvegardeBD
```

### 5-Exercice

En utilisant la commande **awk**, écrire un script affiche les uids et les logins des utilisateurs ayant un compte entre 200 et 400.

Exemple d'affichage :

uid login

200 noure

350 sofia

370 josef

### Rappels :

Les logins et les identifiants des utilisateurs se trouvent dans le fichier **/etc/passwd**, son format :

login:x:uid:gid:commentaire/home/login:/bin/shell

```
Awk -F ":" '{if ($3 > 200 && $3 <400)
    {print $3 "\t" $1}
}' /etc/passwd
```

## 6-Exercice

On souhaite écrire un script en langage VBScript qui réalise les actions suivantes :

-Demander à l'utilisateur un nom de répertoire

-Vérifier l'existence de ce répertoire dans la partition [C:\](#)

-Si le répertoire existe, afficher tout son contenu (fichiers et sous répertoires)

Sinon,

-Créer ce répertoire

-Copier tous les fichiers du répertoire [C:\Temp](#) ayant une extension .7z dans ce nouveau répertoire.

```
rep="C:"
dossier = inputBox ("Entrez le nom d'un dossier")
chemin = rep & dossier
Set objFSO =CreateObject ("Scripting.FileSystemObject")
if objFSO.FolderExists (chemin) = True then
    msgbox "Le dossier existe"
    Set dossier = objFSO.GetFolder(chemin)
    Set listeFichiers = dossier.Files
    Set listeRepertoires = dossier.subFolders
    liste1=" "
    liste2= " "
    For each fic in listeFichiers
        liste1 = liste1 + " - " & fic
    Next
    For Each rep un listeRepertoires
        liste2= liste2 + " - " & rep
    Next
    msgBox liste1,, " Le dossier contient : "
    msgBox liste2,, " Le dossier contient : "
else
    msgBox "Le dossier n'existe pas "
    set nouveau = objFSO.CreateFolder (chemin)
    objFSO.CopyFile "C:\Temp\*.7z ", nouveau
end if
```