

TP 3 : Redirections et Scripts Shell

1 Les redirections

- Concaténer les deux fichiers `/etc/passwd` et `/etc/group` dans le fichier `/tmp/comptes`
- Afficher la liste de tous les processus actifs et sauvegarder le résultat de la commande dans le fichier nommé `psliste`.
- Chercher tous les fichiers de nom contenant "`passwd`" depuis le répertoire `/`. Sauvegarder le résultat de votre commande (sans les messages d'erreurs) dans le fichier `listepasswd`. Tous les messages d'erreur doivent être redirigés dans le fichier spécial `/dev/null`.
- Trier le fichier `/etc/passwd` par ordre alphabétique et enregistrer le résultat dans le fichier `TriPasswd`.
- Trier le fichier `/etc/passwd` suivant l'UID et sauvegarder le résultat à la suite du fichier `TriPasswd` de la question précédente.
- Remplacer la séquence de commandes suivantes par une seule commande :

```
1 cd /etc
2 ls > /tmp/liste.txt
3 cat /tmp/liste.txt
4 wc -l < /tmp/liste.txt
```

- Afficher la liste de tous les processus de l'utilisateur `root`

2 Scripts Shell

- Ecrire le script suivant nommé `Parametres` :

```
1 #!/bin/bash
2
3 # Parametres : Paramètre de position
4
5 echo "Nom du script en cours d'exécution : $0"
6 echo "Nombre de paramètres transmis à un script : $#"
7 echo "Numéro du processus du script en cours d'exécution $0 est : $$"
8
9 echo "Paramètre 1 du script $0 : $1"
10 echo "Paramètre 2 du script $0 : $2"
11 echo "Paramètre 3 du script $0 : $3"
12 echo "Liste de tous les paramètres du script $0 : $*"
13
14 exit 0
```

Une fois le script rendu exécutable, lancez-le de différentes manières :

- Sans paramètre.
 - Avec un seul paramètre.
 - Avec deux paramètres séparés par un espace.
 - Avec trois paramètres séparés par des espaces.
- Créer un script nommé `AfficheRep` qui affiche les caractéristiques et le contenu d'un répertoire dont le nom est donné en paramètre du script.
 - Créer un script `vi2` qui prend en argument un nom de fichier. Le script réalise une sauvegarde du fichier dans le répertoire `/tmp` avant de lancer l'éditeur de textes `vi` pour afficher et/ou modifier son contenu. Le fichier de sauvegarde aura comme nom `nom_parametre_1.numero_processus_script_en_cours`.

4. Lancer le script suivant que vous nommerez *CmdeSet* :

```
1  #!/bin/bash
2
3  # Utilisation de la commande set
4
5  # Première façon : set chaîne_de_caractères
6  set Nom Prenom
7  echo $1
8  echo $2
9
10 # set `commande`
11 set `ls -ld /etc`
12 echo "Résultat : $*"
13 echo "Paramètre 1 de résultat : $1"
14 echo "Paramètre 2 de résultat : $2"
15 echo "Paramètre 3 de résultat : $3"
16 echo "Paramètre 4 de résultat : $4"
17 echo "Paramètre 5 de résultat : $5"
18 echo "Paramètre 6 de résultat : $6"
19 echo "Paramètre 7 de résultat : $7"
20 echo "Paramètre 8 de résultat : $8"
21 Horaire=$8
22
23 # Rôle de la variable IFS
24
25 set $Horaire
26 echo "Paramètre 1 de $Horaire est : $1"
27 echo "Paramètre 2 de $Horaire est : $2"
28
29 # On modifie le séparateur de champs
30 IFS=:
31 echo "Le séparateur de paramètre est : $IFS "
32 set $Horaire
33
34 echo "Paramètre 1 : $1"
35 echo "Paramètre 2 : $2"
36
37 exit 0
```

Que permet la variable IFS ?

5. Créer un script nommé *InfoSys* qui affiche les informations suivantes du système :

- Architecture matérielle.
- Nom de la machine
- Nom du système d'exploitation
- Version du noyau.

La commande *uname* est invoqué une seule fois dans le script. Chaque information est précédée de son libellé ci-dessus.

6. Créer un script nommé *InfoDate* qui affiche la date sous la forme : *nom_du_jour-mois-année* et l'heure sous la forme suivante : il est *xx* heures et *yy* minutes et *zz* secondes.