TP n°4 : Scripts Bash et crontab

Administration Système – ESIEE E3INFO

Ce TP est **à rendre sur elearning** sous forme d'une archive contenant **un script par exercice** (et votre contrab dans un autre fichier).

1 Scripts Bash

Exercice 1

Avec un éditeur de votre choix, créez le script decompte suivant :

```
while test "$1"; do
echo "Le paramètre 1 vaut $1"
shift
done
```

Essayez alors de lancer ./decompte un deux trois quatre après avoir fait un chmod approprié sur le fichier. Quel est l'effet de shift ?

Modifier à présent le script, qui recevra au moins deux paramètres :

- Le premier sera le nom d'un fichier de sortie
- Le deuxième et les suivants seront des répertoires

Votre script devra ajouter une ligne de la forme « Il y a n fichiers dans le répertoire machin ». De plus, s'il rencontre des erreurs (parce que le répertoire n'existe pas ou n'est pas lisible), elles ne devront jamais apparaître à l'écran ; et en lançant votre script deux fois 3 de suite, le fichier de sortie ne devra contenir que les informations relatives à la dernière exécution.

Exemple d'utilisation :

```
$ ./decompte /tmp/sortie /etc /usr
$ cat /tmp/sortie
Il y a 296 fichiers dans /etc
Il y a 12 fichiers dans /usr
$ cat /tmp/sortie
Il y a 6 fichiers dans /home
$
```

Exercice 2

La commande `test x -le y` permet de tester si l'entier x est plus petit ou égal à y. Et la commande test x = y teste si les chaînes de charactères x et y sont identiques. Sachant cela, écrivez un script `sequence.sh [-c t] debut fin` qui écrit tous les nombres entiers entre debut et fin inclus, un nombre par ligne. Si l'option -c est utilisée, alors le script devra afficher les nombres `debut, debut+t, debut+2*t, ...,` sans jamais dépasser fin.

Exercice 3

Exécutez la commande suivante dans un terminal :

```
curl http://cboin.hd.free.fr/info1/fichiers-TD.tgz | tar xvpzf - -C /tmp
```

puis placez-vous dans /tmp/fichiers-TD. Ce répertoire contient des fichiers d'images qui sont toutes au

format JPEG. L'extension des noms de fichiers est parfois correcte (.jpg), mais il arrive aussi qu'elle soit fausse (.tif, .fig, .Z, ...), ou inexistante.

Écrire un script qui corrige ce problème dans le cas général. Il recevra en unique paramètre un répertoire, censé contenir les fichiers à corriger (/tmp/fichiers-TD dans le cas présent). Puis, pour chaque fichier de ce répertoire, écrira la ligne de commande nécessaire (sans la lancer) pour :

- renommer le fichier en .jpg si l'extension n'est pas la bonne
- l'ajouter s'il n'a pas d'extension

S'il n'a rien à faire car l'extension est la bonne, il l'écrira aussi. Aide : pour comparer deux chaînes de caractères entre elles, on peut utiliser la commande test avec l'opérateur =. Et le ! marque la négation (dans la commande test elle-même comme dans une condition en Bash). Comparez, par exemple, les codes de sortie de

```
test abc = abc
test ! xyz = abc
```

Exemple d'utilisation :

```
$ ./rennomer.sh /tmp/fichiers-TD2
Je traite /tmp/fichiers-TD2/crabe 072.png -> mv
/tmp/fichiers-TD2/crabe 072.png /tmp/fichiers-TD2/crabe 072.jpg
Je traite /tmp/fichiers-TD2/image_0030.tif -> mv
/tmp/fichiers-TD2/image_0030.tif /tmp/fichiers-TD2/image_0030.jpg
Je traite /tmp/fichiers-TD2/image_0031.JFIF -> mv
/tmp/fichiers-TD2/image_0031.JFIF /tmp/fichiers-TD2/image_0031.jpg
Je traite /tmp/fichiers-TD2/image_0032.jpg -> est OK, rien à lancer
Je traite /tmp/fichiers-TD2/image_0033.jpg -> est OK, rien à lancer
Je traite /tmp/fichiers-TD2/image_0067.jpg -> est OK, rien à lancer
Je traite /tmp/fichiers-TD2/image_0071.jpg -> est OK, rien à lancer
Je traite /tmp/fichiers-TD2/inconnu.jpg -> est OK, rien à lancer
Je traite /tmp/fichiers-TD2/insecte .bmp -> mv
/tmp/fichiers-TD2/insecte .bmp /tmp/fichiers-TD2/insecte .jpg
Je traite /tmp/fichiers-TD2/Libell -> mv /tmp/fichiers-TD2/Libell
/tmp/fichiers-TD2/Libell.jpg
Je traite /tmp/fichiers-TD2/Une Libellile 010a.Z -> mv /tmp/fichiers-
TD2/Une Libellile 010a.Z /tmp/fichiers-TD2/Une Libellile 010a.jpg
$
```

Exercice 4

La commande `test -d dir` permet de vérifier si dir est un répertoire. Ecrire un script shell qui fera les choses suivantes pour chacun des arguments qu'il recevra :

- Il commencera par vérifier si cet argument est un répertoire. Si ce n'est pas le cas, il affichera « ... n'est pas un répertoire, ignoré. » sur sa sortie d'erreur standard.
- Sinon, deux cas de figure possibles :
  — Soit le répertoire n'est pas vide : le script affichera alors « Le répertoire .. n'est pas vide, ignoré. » sur sa sortie d'erreur standard
  — Soit il l'est : le script affichera alors « Le répertoire ... est vide, voulez-vous le supprimer (o/n) ? » sur sa sortie standard. Si l'utilisateur répond O ou o, alors il supprime le répertoire en question ; sinon, il ne fait rien.

Le script devra sortir avec le code de retour 0 si tous les arguments passés étaient bien des répertoires, sinon avec le code 1 (commande exit).

Exemple d'utilisation :

```
$ mkdir /tmp/vide
$ ./repertoires.sh /etc/passwd /etc/ /tmp/vide
/etc/passwd n'est pas un répertoire, ignoré.
Le répertoire /etc/ n'est pas vide, ignoré
Le répertoire /tmp/vide est vide, voulez-vous le supprimer
   (o/n) ?o
rmdir /tmp/vide
$ echo $?
1
```

Exercice 5

Ecrire un script shell qui recevra un nombre quelconque d'arguments, censés être tous des répertoires. Puis, pour chacun de ces répertoires, fera en parallèle les choses suivantes :

- Il lancera une commande find -type d sur ce répertoire.
- La sortie standard de ce find sera redirigée vers le fichier /tmp/find-xx.txt, et la sortie d'erreur standard vers /tmp/find-xx.log où xx est un entier (pas forcément à 2 chiffres) correspondant au numéro de l'argument correspondant.

Le script devra sortir avec un code de retour égal au nombre de commandes find qui n'ont pas elles-mêmes terminé avec un code de retour nul.

Exemple de sortie :

```
$ ./parafind.sh /etc /var /tmp/vide
J'attends le PID 14784
J'attends le PID 14785
J'attends le PID 14786
$ echo $?
2
$ ls -l /tmp/find-*
-rw-rw-r-- 1 xavier xavier 1028 18 mai 19:56 /tmp/find-1.log
-rw-rw-r-- 1 xavier xavier 90715 18 mai 19:56 /tmp/find-1.txt
-rw-rw-r-- 1 xavier xavier 9861 18 mai 19:56 /tmp/find-2.log
-rw-rw-r-- 1 xavier xavier 147288 18 mai 19:56 /tmp/find-2.txt
-rw-rw-r-- 1 xavier xavier 0 18 mai 19:56 /tmp/find-3.log
-rw-rw-r-- 1 xavier xavier 10 18 mai 19:56 /tmp/find-3.txt
```

2 Quelques exercices avec Crontab

Il est conseillé de réaliser ces planifications dans une machine virtuelle.

1. Programmer un reboot de la machine (dans 5 minutes). Il faudra être utilisateur root. Ne pas oublier d'enlever la tâche après avoir testé son exécution.

2. Ajouter toutes les 5 minutes un message "Bonjour" suivi de la date, dans le fichier /tmp/bonjour.txt
   Vous utiliserez la redirection élémentaire >>

3. Lutilisateur root enregistre de 8h à 18h, uniquement les jours ouvrables :
   - toutes les 2 heures, dans /var/log/proc.txt, tous les processus qui tournent sur la machine
   - tous les 10 minutes, dans /var/log/who.txt, tous les utilisateurs connectés

Ne pas oublier pas d'enlever ces taches du planificateur après les avoir testées.

3 Sauvegarde simple des répertoires personnels (facultatif)

L'objectif est d'effectuer une tâche quotidienne de sauvegarde globale de tous les répertoires personnels présents dans /home sur un serveur FTP sécurisé. Dans un premier temps, on sauvegardera dans le dossier local /var/save

1. Écrire la commande d'archivage compressé de /home/* dans un fichier home.tgz à placer dans /var/save
   Vous utiliserez la commande tar

2. Écrire un script bash permettent l'archivage quotidien. Par exemple, la sauvegarde du 1er mai se fera dans un fichier nommé home.1mai.tgz (vous pouvez utiliser la commande date combinée à la commande set pour générer le nom du fichier)
   https://www.commentcamarche.net/faq/5444-bash-les-parametres#initialiser-des-parametres

3. Automatiser cette tâche à 1h du matin tous les jours avec crontab.

4. Sauvegarder cette archive une fois par semaine le vendredi soir sur un serveur FTP que vous installerez sur une machine virtuelle (proftpd).

5. Programmer la suppression des sauvegardes quotidiennes au bout de 60 jours avec crontab.