



CS230

TP 3 Programmation Shell

Ecriture de scripts de manipulation de fichiers

A rendre dans une semaine à partir d'aujourd'hui :

- Une archive en format compressé (zip, tar.gz, etc), à déposer sur chamillo (partie "Travaux").
- L'archive doit comportée :
 - un compte rendu explicatif en format PDF contenant les noms des binomes, les réponses aux différentes questions, les éventuelles remarques et détails des problèmes rencontrés, les scripts shell et les résultats d'exécutions commentés.
 - les scripts développés

Table des matières

I Manipulation de fichiers	2
II Exercice complet	3

Première partie . Manipulation de fichiers

Exercice 1 : Ecrire un programme shell `ex1.sh` qui compare deux fichiers donnés en paramètre et affiche si sont identiques ou pas (par rapport au contenu).

```
$ ./ex1.sh fich1 fich2
Les fichiers "fich1" et "fich2" sont différents.
```

Indication : vous pouvez utiliser la commande `cmp`.

Exercice 2 : Ecrire un programme shell `ex2.sh` qui cherche sur un fichier l'occurrence d'un mot quelconque donné en paramètre.

```
$ ./ex2.sh fichier MotAChercher
Le fichier "fichier" contient au moins une occurrence du mot "MotAChercher".
$ ./ex2.sh fichier AutreMotAChercher
Le fichier "fichier" ne contient pas le mot "AutreMotAChercher".
```

Exercice 3 : Ecrire un programme shell `ex3.sh` qui supprime tous les fichiers nommés `“.svn”` d'une façon récursive à partir du dossier courant.

```
$ ./ex3.sh
nettoyage des fichiers .svn...
```

Indication : vous pouvez utiliser la commande `find` avec l'option `-exec`

Exercice 4 : Ecrire un programme shell `ex5.sh` qui permet de modifier la date de dernière modification d'un fichier donné en paramètre. Le script devrait afficher la date de dernière modification du fichier puis la modifiera avec l'heure actuelle et enfin réaffichera la date de dernière modification du fichier.

```
$ ./ex4.sh fichier
Avant : -rw-r--r--  1 debbabi  staff   75 Mar 16 21:04 fichier
Après  : -rw-r--r--  1 debbabi  staff   75 Mar 17 11:17 fichier
```

Remarque : le script doit vérifier si le fichier exist déjà.

Exercice 5 : Ecrire un programme shell `ex5.sh` qui compare deux fichiers donnés en paramètre et affiche le plus ancien entre les deux en ce qui concerne la date de modification.

```
$ ./ex5.sh fich1 fich2
"fich1" est plus ancien que "fich2".
```

Exercice 6 : Ecrire un programme shell `ex6.sh` qui crée un répertoire nommé `“Documents”` contenant 10 fichiers nommés `“Doc1.txt”` à `“Doc10.txt”`. Chaque fichier contient une seule ligne (comme sur l'exemple ci-dessous).

```
$ ./ex6.sh
$ cat Documents/Doc1.txt
"Doc1" est le premier document.
$ cat Documents/Doc2.txt
"Doc2" est le deuxième document.
$ cat Documents/Doc10.txt
"Doc10" est le dixième document.
```

Deuxième partie . Exercice complet

Exercice 7 : Ecrire un programme shell `sauvegarde.sh` qui fait une sauvegarde du répertoire racine de l'utilisateur (`HOME`). Le script doit avoir les fonctionnalités suivantes :

- Le script doit accepter au maximum un paramètre qui désigne le dossier destination. Si aucun paramètre est présent, le script utilise un dossier pré-déterminé dans une variable d'environnement `BACKUP_FOLDER` (à rajouter automatiquement dans le fichier `.bash_profile` ou `.bashrc` au premier lancement du script si elle n'existe pas déjà). Si plusieurs paramètres sont présents, sortir après avoir affiché un message d'utilisation du script.
- Le script doit créer un seul fichier d'archive compressé du dossier racine avec le modèle de nom suivant : `NomUtilisateur-02042014-0934.tar.gz` où `NomUtilisateur` est le nom de l'utilisateur en cours, `02042014` correspond à la date courante (dans l'exemple 02 avril 2014), et `0934` correspond à l'heure actuelle lors du lancement du script (dans l'exemple 9h et 34 minutes).
- S'il existe déjà un fichier de sauvegarde dans le dossier destination qui date d'au moins `BACKUP_INTERVAL` jour(s) ; demander à l'utilisateur une confirmation avant de lancer le processus de sauvegarde. La variable d'environnement `BACKUP_INTERVAL` est à rajouter automatiquement dans le fichier `.bashrc` ou `.bash_profile` au premier lancement du script s'il elle n'existe pas déjà.
- Déterminer si le répertoire de destination a assez d'espace libre pour contenir la sauvegarde.
- Afficher un message indiquant à l'utilisateur le nom, la taille et le dossier où se trouve la sauvegarde nouvellement créée.