

TP n° 2 : Scripts shell bash

Exercice 0

Tester les exemples vus en cours

Exercice 1

Créer un script `test-fichier`, qui à partir d'un nom de fichier ou de répertoire indique s'il s'agit d'un fichier ou d'un répertoire et précise les permissions d'accès pour l'utilisateur qui a lancé la commande

Exemple de résultats :

```
/etc est un répertoire lisible par toto
```

```
/etc/passwd est un fichier ordinaire lisible modifiable exécutable par toto
```

Exercice 2 - Afficher le contenu d'un répertoire

Écrire un script `listedir.sh` permettant d'afficher le contenu d'un répertoire en séparant les fichiers et les répertoires.

Exemple d'utilisation :

```
$ ./listdir.sh /etc/rc.d
```

affichera

```
-----  
Les fichiers de /etc/rc.d sont :  
fic1  
fic2  
toto  
-----  
Les répertoires de /etc/rc.d sont :  
rep1  
test
```

Remarque : positionner la variable d'environnement `IFS=$'\n'` permet de gérer les cas où les noms de fichiers contiennent des espaces.

Exercice 3 : Lister les utilisateurs

Écrire un script bash affichant la liste des noms de login des utilisateurs définis dans `/etc/passwd` ayant un UID supérieur à 500.

Exercice 4 : lecture au clavier

La commande bash `read` permet de lire une chaîne au clavier et de l'affecter à une variable. exemple :

```
echo -n "Entrer votre nom: "  
read nom  
echo "Votre nom est $nom"
```

Écrire un script qui propose à l'utilisateur de visualiser page par page chaque fichier texte du répertoire spécifié en argument. (un fichier texte sera un fichier terminant par l'extension `txt`) Le script affichera pour

chaque fichier texte (et seulement ceux là) la question "voulez vous visualiser le fichier machintruc ?". En cas de réponse positive, il lancera `more`, avant de passer à l'examen du fichier suivant.

Exercice 5 : itération, chaînes de caractères, expressions

La variable spéciale de bash `$RANDOM` permet de générer un nombre entier aléatoire entre 0 et 32767.

1. Écrire un script qui crée 10 fichiers dont le nom est un nombre pris au hasard entre 0 et 100 auquel on ajoute l'extension `.txt`
2. Modifier le script pour avoir 2 options :
 - (option `-c`) créer les fichiers au hasard comme ci-dessus (on passe en paramètre le nombre de fichiers créés)
Exemple : `./exo5 -c 10`
 - (option `-s`) supprimer tous les fichiers du répertoire courant de la forme `<un nombre>.txt`
Exemple : `./exo5 -s`

Exercice 6 : ps-perso

On souhaite écrire un script bash permettant de recenser l'ensemble des processus en cours d'exécution. Le système crée un répertoire qui porte comme nom un nombre (qui est le pid du processus) dans `/proc` pour chaque processus en cours d'exécution.

1. Écrire un script permettant d'afficher tous les pid des processus en cours d'exécution. (sans utiliser la commande `ps` bien sûr)
Aide : pour repérer si une chaîne est un nombre on peut faire le test `$((nom))=0` qui renvoie vrai si `nom` ne représente pas un nombre entier
2. Modifier le script pour qu'il affiche tous les pid des processus compris entre 2 valeurs passées en paramètres.
3. Modifier le script pour qu'il affiche pour chaque pid trouvé le nom du processus correspondant
Aide : le fichier `status` contient une ligne qui contient le nom du processus
On obtiendra un affichage du style suivant :
`pid=818, nom = getty`
`pid=845, nom = atd`
`...`

Exercice 7 : tueur de xclocks

On souhaite écrire un script bash permettant de tuer tous les processus dont le nom est `xclock`.