

TP N°7: SCRIPTS

NETTOYER.SH

1) Créer les scripts "outils" ci-dessous.

Scripts destinés à être appelés par d'autres scripts pour déléguer une partie du traitement.

Scripts robustes, avec traces, messages, compte-rendus...

Scripts de contrôles d'arguments :

Les 3 scripts ci-dessous sont appelés par un script en lui passant comme arguments ceux qui lui ont été fournis pour vérifier leur conformité.

Pour le TP, on considère que ces 3 scripts seront impérativement **appelés via la commande source**.

```
#####
# argumentUnique.sh : script utilisé par un autre script pour savoir
# s'il a bien été appelé en lui passant un seul argument
#####
OK=0
ERR_ARGUMENT_ABSENT=1
ERR_ARGUMENTS_EN_TROP=2
ERR_ARGUMENT_INEXISTANT=3
```

Les 2 scripts ci-dessous appellent le script ci-dessus.

```
#####
# argumentFichier.sh : script utilisé par un autre script pour savoir
# s'il a bien été appelé en lui passant un unique fichier lisible
#####
OK=0
ERR_ARGUMENT_ABSENT=1
ERR_ARGUMENTS_EN_TROP=2
ERR_ARGUMENT_INEXISTANT=3
ERR_ARGUMENT_NON_FICHER=4
ERR_FICHER_NON_LISIBLE=5
```

```
#####
# argumentRepertoire.sh : script utilisé par un autre script pour savoir
# s'il a bien été appelé en lui passant un unique répertoire lisible
#####
OK=0
ERR_ARGUMENT_ABSENT=1
ERR_ARGUMENTS_EN_TROP=2
ERR_ARGUMENT_INEXISTANT=3
ERR_ARGUMENT_NON_REPERTOIRE=4
ERR_REPERTOIRE_NON_LISIBLE=5
ERR_REPERTOIRE_VIDE=6
```

Pour le TP, on considère que les scripts suivant seront impérativement **appelés en créant un shell fils**.

2) Créer le script **choisirFichier.sh**

```
#####
# choisirFichier.sh propose un menu pour choisir parmi les 5 plus gros fichiers
# du répertoire fourni en paramètre (le 0 permet d'annuler le choix)
# Ce script affiche le nom du fichier choisi et retourne un numéro de fichier entre 1 et 5
# (le 1 correspond au plus gros fichier du répertoire , le 2 au deuxième plus gros...)
#####
ANNULATION=0
ERR_REP=6
```

3) Dans un fichier **nettoyer.sh**, créer une fonction **supprimer()** qui supprime le fichier qui lui est passé en paramètre, avec demande de confirmation avant la suppression

Créer le script **nettoyer.sh** :

```
#####
# nettoyer.sh propose de supprimer un des 5 fichiers les plus gros du répertoire fourni en
paramètre
#####
OK=0
ERR_ARGUMENT_ABSENT=1
ERR_ARGUMENTS_EN_TROP=2
ERR_ARGUMENT_INEXISTANT=3
ERR_ARGUMENT_NON_REPERTOIRE=4
ERR_REPERTOIRE_NON_LISIBLE=5
ERR_REPERTOIRE_VIDE=6
ERR_FICHIER_SUPPRESSION=7
ERR_SAISIE_REPONSE=8
```

FORMAT.CSV.SH

Pour le script suivant créer une fonction **ctrl_extension()** qui contrôle la conformité de l'extension et une fonction **ctrl_format()** pour contrôle la conformité de son contenu

```
#####
# Le script format.csv.sh vérifie que l'argument fourni est bien un fichier lisible
# au format csv:
# son nom porte une extension csv
# chacune des lignes compte un même nombre de champs séparés par des ","
#####
OK=0
ERR_ARGT=1
ERR_EXT=2
ERR_FORMAT=3
```

UTILISATEURS.SH

Utiliser les opérateurs sur les chaînes $\${...}$, pas les filtres ...

Ecrire un script ***utilisateurs.sh*** qui, à partir de la lecture du fichier `/etc /passwd`, affiche ligne par ligne le nom des comptes déclarés avec en regard soit :

- Shell : si c'est un compte qui lance un shell,
- Pas de Shell : si c'est un compte ne qui lance pas de shell (`/bin/false` ou `/usr/sbin/nologin`),
- l'exécutable lancé, sans le chemin, sinon.

Exemple de trace `./utilisateurs.sh`

```
root=>Shell
daemon=>Shell
bin=>Shell
sync=>sync
games=>Shell
```