

Scripts Bash — Révisions

Avril 2019

Exercice 1

À l'aide de la commande `date`, écrire un script nommé `exo1.sh` qui génère un fichier nommé d'après la date courante :

- format du nom : `fichier-133742.txt` , s'il était 13 h 37 min 42 s à l'exécution du script
- contenu du fichier : `vendredi 19 avril 2019, 13:37:42 (UTC+0200)`

Évidemment, à chaque appel du script, un nouveau fichier, avec un nouveau contenu, sera créé.

Exercice 2

Écrire un script shell bash nommé `exo2.sh` qui :

- prend un nom de répertoire comme unique paramètre. S'il y a plus ou moins d'un paramètre, vous devrez afficher le message «trop de paramètres ou pas assez de paramètres». Si le paramètre passé n'est pas un répertoire, vous devrez afficher «le paramètre passé n'est pas un nom de répertoire valide».
- affiche le nombre de fichiers de ce répertoire qui sont exécutables par l'utilisateur qui lance le script. Exemple : la commande `exo2 machin` exécutée par toto affiche «le répertoire machin contient 54 fichiers executables par toto».

Exercice 3

Écrire un script shell bash nommé `exo3.sh` qui :

- crée un répertoire `util` dans `$HOME` s'il n'existe pas. S'il existe déjà un message d'erreur doit s'afficher et le programme doit s'arrêter.
- récupère les noms des utilisateurs du système dans le fichier `/etc/passwd` (dont le 1^{er} champ est le nom de l'utilisateur)
- crée un fichier par nom obtenu à l'étape précédente dans le répertoire `$HOME/util` . Les fichiers porteront le nom des utilisateurs et contiendront 2 lignes : la première contiendra l'id de l'utilisateur et la seconde le nom du shell utilisé par l'utilisateur.

Exemple : si `/etc/passwd` contient 3 utilisateurs `u1` , `u2` et `u3` : seront créés 3 fichiers `$HOME/util/u1` ; `$HOME/util/u2` et `$HOME/util/u3`.