

TP n° 3 - Scripts Shell

Pour chacune des tâches suivantes, il s'agit de trouver une séquence de commandes shell la réalisant. Si nécessaire, on n'hésitera pas à :

- utiliser des fichiers temporaires pour stocker des données intermédiaires
- utiliser des variables pour stocker le résultat d'une commande avec la syntaxe `var=$(commande)`
- faire des opérations arithmétiques avec la syntaxe `=$((i + 1))` ou `$(i + 1)`
- utiliser des boucles `for` (avec `seq`) ou `while`
- écrire les commandes dans un script...

Remarque : des commandes utiles sont données en indication, on prendra le temps d'en lire le manuel.

Pour cette séance, créer le répertoire `TP3`, et placez-vous dans `dataTP3` pour la suite.

Exercice 1. Écrire un script shell qui tire au hasard un mot dans le dictionnaire du système `/usr/share/dict/words`, à l'aide de `wc` et de la variable `$RANDOM` qui contient à chaque utilisation un entier aléatoire. Le mot est ensuite affiché.

Exercice 2. Écrire un script qui affiche les termes de la suite de Syracuse, définie par :

$$u(n+1) = u(n) / 2 \text{ si } u(n) \text{ est pair et } 3 * u(n) + 1 \text{ sinon.}$$

partant de `u(0) = 42`. On s'arrête lorsque la valeur 1 est atteinte.

Exercice 3. Écrire un script qui convertit en décimal un nombre binaire contenu dans la variable `$binaire` (initialiser par exemple `binaire` à `110110`).

Exercice 4. Écrire un script qui affiche à l'écran toutes les lignes paires du fichier `/etc/passwd`.

Exercice 5. Soit un fichier dont le nom est dans une variable `fnom` (vous pouvez par exemple, mettre dans la variable `fnom` le nom d'un fichier script précédemment écrit). Écrire un script shell qui crée un fichier dont le nom est le contenu de la variable `fnom` auquel on a ajouté le suffixe `2` (par exemple, si le fichier s'appelle `foo`, le fichier qu'on doit créer s'appellera `foo2`) et tel que le contenu de ce fichier créé est le même que celui du fichier d'origine sans aucune ligne vide.

Pour les exercices suivants, télécharger l'archive `dataTP3.zip`, la décompresser et se placer dans le répertoire `dataTP3` pour la suite.

Exercice 6. Écrire un script shell qui forme un juron du capitaine Haddock en tirant au hasard un adjectif et un qualificatif dans les fichiers `haddock/noms` et `haddock/qualificatifs`.

Exercice 7. Écrire un script qui crée dans le même répertoire un fichier `cyrano_num` obtenu à partir du fichier `txt/cyrano.txt`, en numérotant ses lignes à l'aide de `seq` et `paste`. On notera au passage que `cat -n` permet d'en faire autant plus simplement.

Exercice 8. Écrire un script qui met les 3 premières lignes des fichiers texte du répertoire `haddock` dans un fichier `resume.txt`. Le fichier `resume.txt` contiendra pour chaque fichier texte, son nom et ses 3 premières lignes.

Exercice 9. Écrire un script qui affiche pour chacun des fichiers texte du répertoire `data/haddock` les informations : nom, nombre de lignes, nombre de caractères, le nom du propriétaire.

Exercice 10. Écrire un script qui affiche des informations de chaque fichier du répertoire `data/haddock` dans un langage plus convivial, par exemple "Le propriétaire a les droits : lire, écrire". Indication : utiliser `ls -l`, `cut -f` et `cut -c`.