

TP n°3 Boucles, tests et arithmétique

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.

Le répertoire `data` est obtenu à partir du fichier `data.tar.gz` sur Celene (comme pour TP2).

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

Exercice 3. Écrire un script qui numérote les lignes du fichier `data/txt/cyrano.txt`, à l'aide de `seq` et `paste`. On notera au passage que `cat -n` permet d'en faire autant plus simplement.

Exercice 4. 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 5. Écrire un script qui convertit en décimal un nombre binaire contenu dans la variable `$binaire` (initialiser par exemple binaire à 110110).

Exercice 6. Écrire un script qui met les 3 premières lignes des fichiers texte du répertoire `data/haddock` dans un fichier `resume.txt`. Le fichier `resume.txt` contiendra pour chaque fichier texte, son nom et ses 3 premières lignes.

Exercice 7. Écrire un script qui affiche à l'écran toutes les lignes paires du fichier `/etc/passwd`.

Exercice 8. 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 9. 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`.

Exercice 10. La chasse au trésor
Inutile d'essayer de sauter des étapes...

1. Afficher le contenu du fichier `tp3/sheep/sheep` sur 28 colonnes (par exemple à l'aide de `pr`).
2. Suivre la chaîne de liens initiée dans le fichier pointé par la réponse précédente.

3. Nettoyer le fichier de fin de chaîne des caractères non alphanumériques pour découvrir une nouvelle indication.
4. Suivre la nouvelle chaîne. Attention aux pièges !
5. Déterminer le fichier pointé par le fichier de fin de la chaîne précédente (à l'aide de **shasum**).
6. Analyser ce fichier (à l'aide de **file**). Le déstructurer autant de fois que nécessaire (à l'aide de **zcat** et **bzcat**) pour en tirer quelque chose de lisible.
7. The rest is silence (Hamlet V.2).