
TP n° 1: Introduction au Shell (corrigé)

Exercice 1.*Shell we start ?*

Le *shell* est un langage de programmation interprété : il est donc possible d'écrire des programmes plus complexes qu'une simple ligne.



Pour la facilité de lecture et de réutilisation, les programmes sont souvent écrits dans un fichier. Généralement, la première ligne d'un script *shell* indique le nom du programme qui doit l'interpréter à travers la syntaxe

```
#!/chemin/vers/le/programme
```






Par exemple `#!/bin/sh`.

Ainsi, à titre d'illustration, voici le Hello World en shell :


```
#!/bin/sh
echo "Hello, World!"
```

1. Que font les commandes suivantes : `man`, `cd`, `ls`, `pwd`, `mkdir`, `rm`, `mv`, `cp`, `file`, `echo`, `cat`, `head`, `tail`, `touch`, `history`, `cal`, `grep` ? En utilisant la commande `type`, déterminer si les commandes précédentes sont des programmes ou des fonctions du shell.  Utiliser la commande `man` pour déterminer ce que fait chaque commande.
2. Écrire une boucle affichant les entiers divisibles par 3 entre 1 et 100 (utiliser un `if`). 

```
#!/bin/bash
for entier in {1..100}
do
    if [ $(( ${entier} % 3 )) -eq "0" ]; then
        echo $entier
    fi
done
```

3. Tester le "hello world" : copiez le contenu dans un fichier `hello` ; donnez-lui les permissions d'exécutables ; exécutez `hello`.  Utiliser la commande `chmod +x ./hello.sh` pour ajouter la permission exécutable.
4. La commande `wc` permet de compter le nombre de mots d'un fichier. L'option `-l` compte à la place le nombre de lignes. Lancez la commande `ls | wc -l`. À quoi correspond le résultat ?
Le caractère `|` (pipe, ou tuyau) prend la sortie de la commande à sa gauche, et la donne en entrée à la commande à sa droite. À l'aide de `grep`, `cat` et du pipe, afficher le modèle du CPU (regarder le fichier `/proc/cpuinfo`).
 La première commande affiche le nombre de fichiers et de dossiers dans le répertoire courant.
Pour afficher le modèle du CPU, utiliser `cat /proc/cpuinfo | grep "model name"`. Si une ligne apparaît plusieurs fois (par exemple 4 fois), cela veut dire que votre processeur à 4 coeurs / threads.
5. Modifier le script pour qu'il affiche son premier argument.  `echo "$1"` // Les guillemets sont importants
6. Modifier le script pour qu'il affiche **tous** ses arguments.  `echo "$@"`
7. Écrire un script shell qui affiche tous ses paramètres, mais **un** par ligne.
 `for I in "$@"; do echo "$I"; done`
8. Écrire un script shell qui prend en paramètres deux noms de fichiers : `file1` `file2` et qui supprime `file2` seulement si `file1` existe.

Attention : `file1` et `file2` doivent être des fichiers, et non pas des répertoires.

```
 if [ -f "$1" ] ; then rm "$2" ; fi
ou
if test -f "$1" ; then rm "$2" ; fi
```

Exercice 2.

Un peu de géographie

1. Télécharger le fichier `http://perso.ens-lyon.fr/etienne.moutot/static/teach/2018-2019/asr2/OSMTowns.txt` contenant les coordonnées GPS des villes de France, en utilisant la commande `wget` ou `curl`.

Utiliser `wget "http://perso.ens-lyon.fr/etienne.moutot/static/teach/2018-2019/asr2/OSMTowns.txt"` ou `curl "http://perso.ens-lyon.fr/etienne.moutot/static/teach/2018-2019/asr2/OSMTowns.txt" > OSMTowns.txt`

2. Compter le nombre de villes. `wc -l OSMTowns.txt`
3. Trouver la ville la plus à l'Ouest de la France et la ville la plus à l'Est de la France. `LANG=EN sort OSMTowns.txt --field-separator=$'\t' --general-numeric-sort --key=2|head -n 1`

Cette question est plus difficile qu'il n'y paraît. Une commande qui fonctionne est par exemple :

```
LANG=EN sort OSMTowns.txt --field-separator=$'\t' --general-numeric-sort --key=2|head -n 1
```

Explication :

- On préfixe la commande de `LANG=EN`, ce qui permet de changer la langue utilisée par `sort` pour trier les nombres. En français, en effet, les nombres flottants utilisent une virgule, alors qu'ils utilisent un point en anglais.
- On spécifie à `sort` que le séparateur entre les différentes colonnes du fichier est une tabulation (caractère `'\t'`), mais comme le caractère tabulation est un caractère "spécial", il faut utiliser un dollar. On aurait pu aussi simplement utiliser `--field-separator=" "` (avec le caractère tabulation entre les double quotes).
- On précise avec `--general-numeric-sort` que l'on souhaite trier selon l'ordre numérique, et non selon un ordre alphanumérique (où `z` est avant `A` par exemple).
- On spécifie que l'on trie selon la seconde clef avec `--key=2` - Enfin, on récupère la première ligne avec la commande `head` (ou la dernière, en utilisant `tail`).

4. En utilisant ce fichier, écrire un script shell `coordonnees` qui prend en argument le nom d'une ville et qui renvoie ses coordonnées GPS. ¹

```
#!/bin/bash
LATLONG=$(grep "^$1" OSMTowns.txt|head -n1)
# ^ tabulation character
# We use a head as the output might contain multiple city
# named the same way.
LAT=$(echo "$LATLONG" |cut --delimiter " " --fields=3)
LONG=$(echo "$LATLONG" |cut --delimiter " " --fields=2)
echo "La ville $1 a pour latitude $LAT et longitude $LONG"
```

5. Toujours en utilisant ce fichier, écrire un script shell qui prend en argument le nom d'une ville et renvoie un lien du type `https://www.openstreetmap.org/#map=18/45.72980/4.82712` (en remplaçant `45.72980` par la latitude de la ville et `4.82712` par sa longitude). Il suffit de modifier la ligne "echo" dans la réponse précédente.
6. Utiliser la commande `grep` pour afficher l'ensemble des villes contenant "chien" dans leur nom. En une seule utilisation de `grep`, trouver l'ensemble des villes contenant "Chat" ou "chat". `grep "chien" OSMTowns.txt`, puis `grep "\(chat\|Chat)" OSMTowns.txt` (ou `grep -i "chat" OSMTowns.txt`)

Exercice 3.

Curling. - Difficile

Le but de cet exercice est d'écrire un script qui prendra en paramètre le nom d'une ville quelconque, effectuera une recherche sur Wikipedia pour trouver une image correspondant à cette ville, puis trouvera la distance entre Lyon et la ville donnée. Ensuite il téléchargera l'image et rajoutera le nom de la ville et la distance à Lyon sur l'image.

Les étapes à suivre :

1. Télécharger le fichier `http://perso.ens-lyon.fr/etienne.moutot/static/teach/2018-2019/asr2/gps-dist.py` qui est un programme permettant de calculer la distance entre deux points donnés par leurs coordonnées GPS. En utilisant ce programme, écrire un script shell `distance` qui prend en argument le nom d'une ville et qui renvoie sa distance à Lyon.

Exemple :

1. Il est possible d'utiliser la commande `cut` pour découper un fichier selon des délimiteurs. Par exemple, `echo "A,B,C,D,E" | cut --delimiter="," --fields 3` renvoie `C`

```
$ ./distance "Porcelette"  
496
```

Les fichiers de correction sont disponibles à l'adresse <http://perso.ens-lyon.fr/etienne.moutot/static/teach/2018-2019/asr2/tpl/>

2. En utilisant *Mediawiki API*², écrire un script `image` pour afficher le lien de la première image correspondant à cette ville.

```
$ ./image "Porcelette"  
https://upload.wikimedia.org/wikipedia/commons/2/24/Vue\_du\_village\_de\_Porcelette.JPG
```

3. En combinant les deux scripts précédents, écrire un troisième script `descr` qui prend en argument un nom de ville et crée une image qui contient l'image de la ville avec en dessous un texte indiquant sa distance à Lyon dans un fichier `image.jpg`. Par exemple `./descr "Porcelette"` crée l'image suivante



4. (Optionnel) Modifier le script `image` pour prendre un entier supplémentaire pour sélectionner non pas la première image dans les résultat mais les n premières images.
Utiliser l'API de commons et non celle de wikipédia : <https://commons.wikimedia.org/w/api.php>
5. (Optionnelle) Écrire un script `collage` qui prend en argument n^2 images et qui réalise un collage sous forme de mosaïque.
(a) (Optionnelle) Rajouter une description sous les image (par exemple le *nom de base* de l'image).
6. (Optionnelle) Utiliser les scripts précédents pour écrire un script qui prend un mot au hasard (`/usr/share/dict/*`) et réalise un collage avec les images issues de Wikipedia.

2. <https://fr.wikipedia.org/w/api.php>