



TP 4 - Scripts - groupe FA

Préambule

Pour réaliser ce TP, il faut écrire des scripts shell. Vous devez utiliser un éditeur de fichier texte (comme gedit) pour créer un nouveau fichier, par exemple le fichier `mon_script.sh`. La première ligne de votre fichier de script shell doit être :

```
#!/bin/bash
```

Vous pouvez ensuite écrire des commandes, des tests ou des boucles.

Pour ce TP, vous aurez besoin de connaître les variables shell et commandes suivantes :

- `$#` : pour récupérer le nombre d'arguments sur la ligne de commande ;
- `$1` et `$2` : correspondent respectivement aux premier et deuxième argument de la ligne de commande, tandis que `$0` correspond au nom du script qui s'exécute.
- `exit X` permet de terminer le processus avec le code de retour `X`, où `X` est un entier.
- `wc` permet de compter le nombre de caractères, mots et lignes d'un fichier ;
- `expr` permet de faire des calculs sur une variable ;
- `head` et `tail` permettent respectivement de récupérer les premières et dernières lignes d'un fichier ;

Et bien sûr :

- `ls`, `cat` et `man`.

Pour réaliser une comparaison ou un test, on peut utiliser l'opérateur `[[...]]`.

Test	Signification
<code>[[-e fic]]</code>	le fichier <code>fic</code> existe
<code>[[-r fic]]</code>	le fichier <code>fic</code> est lisible
<code>[[-w fic]]</code>	le fichier <code>fic</code> écrivable
<code>[[-x fic]]</code>	le fichier <code>fic</code> exécutable
<code>[[-f fic]]</code>	<code>fic</code> est un fichier
<code>[[-d dir]]</code>	<code>dir</code> est un dossier
<code>[[\$a -gt \$b]]</code>	<code>a</code> est plus grand que <code>b</code>
<code>[[\$a -ge \$b]]</code>	<code>a</code> est plus grand ou égal à <code>b</code>
<code>[[\$a -lt \$b]]</code>	<code>a</code> est plus petit que <code>b</code>
<code>[[\$a -le \$b]]</code>	<code>a</code> est plus petit ou égal à <code>b</code>
<code>[[\$a -eq \$b]]</code>	<code>a</code> est égal à <code>b</code>
<code>[[\$a -ne \$b]]</code>	<code>a</code> est différent de <code>b</code>

Question 1 : Arguments de la ligne de commande

- Écrire un script qui permet d'afficher les trois premiers arguments de la ligne de commande.
- Modifier votre script pour qu'il vérifie qu'il y a exactement 3 arguments sur la ligne de commande. S'il n'y a pas 3 arguments, vous pouvez afficher un message pour expliquer qu'il faut exactement 3 arguments.

Question 2 : Maximum

1. Écrire un script `max.sh` prenant en paramètre 2 nombres et affichant le plus grand d'entre eux
2. Modifier votre script pour qu'il prenne 3 nombres en paramètre et qu'il soit capable de trouver le maximum de ces 3 nombres.

Question 3 : Lisibilité Écrire le script `est_lisible.sh` permettant de tester si un fichier dont le nom est passé en argument existe et est lisible par l'utilisateur.

Question 4 : Affichage massif Écrire le script `bigcat` permettant d'afficher à l'écran le contenu d'une série de fichiers. Avant chaque contenu le script affichera le nom du fichier. Le message `Inaccessible` sera affiché si un fichier est inexistant ou ne peut être lu.

Question 5 : Gestion des types de fichiers

1. Écrire un script qui prend un argument indique si le paramètre est un fichier, un répertoire ou autre chose.
2. Testez également si le fichier est un script (en utilisant les commandes `file` et `grep`)
3. Modifiez le script pour qu'il teste tous les fichiers présents dans le dossier courant et ses sous-dossiers.

Question 6 : Pour aller plus loin

- Comme pour la question 1, écrivez un script qui permet de trouver le maximum sur 6 nombres passés en paramètre.
- Comme pour la question 1, mais avec n nombres passés en paramètre, avec $n \geq 2$.