

TP5 - Les scripts et les jobs

Inès de Courchelle

2019 - 2020

1 Quelques conseils !

1.1 Quote, quote-quote, ou back-quote

Dans un shell, tapez les commandes suivantes :

```
ls -l
ls -l | wc -l
echo "ls -l | wc -l"
echo 'ls -l | wc -l'
echo `ls -l | wc -l` (back-cote combinaisons touches : altGr + 7)
```

La première des écriture des commandes `echo` est écrite avec des "doubles-quotes"¹. La seconde est écrite avec des simples quotes², la dernière est écrite avec des back-quotes. Notez bien la différence qui existe entre les deux dernières commandes. Cette différence est utile lorsque l'on veut mixer des commandes dans d'autres.

2 Exercices

2.1 Mon premier script

1. Recopiez le script suivant appelez `script1.sh` dans un éditeur de texte soit `vi` ou `emacs`, et exécutez le.

```
1 #!/bin/sh
2 for i in `seq 1 10`
3 # on peut aussi faire for i in $(seq 1 10)
4 do
5     echo "Rock n roll"
6 done
```

-
1. Appelées aussi quote-quote
 2. Nomées quote

Réponses

2. Que fait le script suivant ? Décrivez le programme ligne par ligne.

```
1 #!/bin/sh
2 i1=`ls -a`
3 i2="grep -R 'tutu' ~"
4 echo $i1
5 echo $i2
```

Réponses

3. Exécutez un shell, qui affiche "bonjour" directement dans le terminal sans passer par un éditeur de texte.

Réponses

2.2 Mon deuxième script

Ecrivez un programme, appelé `petitScript.sh` qui prend 5 paramètres/arguments et qui affiche :

- Son nom
- Son nombre de paramètres/arguments
- Le numéro de processus courant (celui qui exécute le script shell)
- Sa liste de paramètres que vous lui avez passés

Quelques indices de variables utiles :

- `$0` ⇒ contient le nom du programme en cours
- `$1..$9` ⇒ contiennent les différentes valeurs des arguments/paramètres
- `$#` ⇒ contient le nombre de paramètres/arguments
- `$*` ⇒ contient une liste de l'ensemble des paramètres/arguments
- `$?` ⇒ contient l'état de sortie de la dernière commande. Si la commande s'est bien passée, alors elle renvoie 0, sinon une valeur différente de zéro.
- `$$` ⇒ contient le numéro du processus en cours

Réponses

2.3 Mon troisième script

1. Créez dans votre répertoire de connexion un répertoire `monTroisieme` qui contient les fichiers suivants (qui peuvent être vides) :
 - `p1.c`
 - `p2.pas`
 - `p3.py`
 - `memo`
 - `p4.c`
 - `p5.txt`
 - `p6.sh`
 - `p7.sh`
2. Rajoutez manuellement le droit d'exécution à quelques fichiers, par exemple à : `p3.py`, `p6.sh` et `p7.sh`
3. Créez dans votre répertoire de connexion un script `maTache.sh`, qui doit :
 - être appelé avec un et un seul paramètre/argument
 - vérifier que le paramètre/argument est bien un répertoire
4. Ajoutez au script `maTache.sh` la capacité de consulter tous les fichiers du répertoire passé en paramètre et affiche le nom de ceux qui sont exécutables. Pour tester, passez en paramètre le répertoire `monTroisieme`.
Attention un fichier en unix est dit exécutable s'il possède les droits d'exécution.
5. Modifiez le script pour qu'il affiche la taille de chacun des fichiers exécutables.
6. Modifiez le script pour qu'il compte le nombre des fichiers exécutables
7. Modifiez le script pour créer un répertoire `exec` et déplacer l'ensemble des exécutables à l'intérieur.

2.4 Les modes d'exécution des processus

1. Créez un "script" qui effectue une boucle infinie contenant l'instruction `sleep`. indices : consulter le manuel de `sleep`

Réponses

2. Exécutez le script en mode foreground. Pouvez vous exécuter encore des commandes dans le terminal dans lequel vous avez lancé le script ?

Réponses

3. Arrêtez le script, puis vérifiez que le script n'est plus dans la liste des processus en cours.

Réponses

4. Exécutez le script de manière à pouvoir utiliser le terminal.

Réponses

5. Exécutez le script en mode foreground puis passez le processus en mode background.

Réponses

6. Exécutez en background le script, puis recherchez le dans la liste des processus en cours, et enfin tuez le.

Réponses

7. Exécutez 2 script d'affiler en mode background (&), et tuez les deux processus.

Réponses

3 Et si je révisai !

1. Nous considérons le fichier de notes suivant :

```
Liste de notes EISTI 2017
paco 10
pepe 5
jose 18
juan 15
luis 19.5
toto 2
titi 5
```

Créez un script qui permet un affichage trié d'un fichier de note sans trier l'entête (première ligne)

Réponses

2. Nous considérons un script `toto.sh` :

(a) Affichez la liste des jobs dont vous êtes le propriétaires

Réponses

(b) Affichez dans le terminal l'identifiant local des processus lancés

Réponses

(c) Exécutez le processus

Réponses

(d) Passez le processus en mode background

Réponses

(e) Arrêtez le processus `toto.sh`

Réponses