

Exercices TP

Recopiez le contenu du répertoire `/Infos/lmd/2011/licence/ue/li219-2012fev/TP3` qui contient tous les fichiers nécessaires pour la réalisation de ce TP et en particulier le fichier `compte_rendu_TP3.txt` que vous devez compléter au fur et à mesure et soumettre à la fin de la séance.

Exercice 23 – Passage de paramètres et exportation de variables

Question 1

Écrivez un script `pere_para.sh` qui prend en paramètre un entier et crée un nombre égal à cet entier de processus fils, en transmettant **en paramètres** à chacun sa position dans l'ordre de création et le nombre total de fils qui doivent être créés.

Écrivez le script `fils_para.sh` exécuté par chaque processus fils et qui affiche le numéro (i.e. la position du processus) et le nombre total de processus créés. Voici un exemple d'exécution :

```
Prompt% ./pere_para.sh 3
```

```
Je suis le fils 1 sur 3
Je suis le fils 2 sur 3
Je suis le fils 3 sur 3
```

Question 2

Nous souhaitons maintenant avoir deux nouveaux scripts `pere.sh` et `fils.sh` avec le même comportement que dans la question précédente mais le père ne doit transmettre **aucun paramètre** à ses fils. Écrivez les scripts `pere.sh` et `fils.sh`.

Exercice 24 – Hiérarchie de processus `shell`

Dans cet exercice vous n'avez pas de script à écrire. Toutes les commandes sont à saisir directement dans une fenêtre terminal.

Question 1

Ouvrez un terminal, initialisez à 30 une variable `A`, affichez la valeur de `A`, affichez le PID du `shell` en train de s'exécuter.

Question 2

Depuis le terminal ouvert à la question précédente, lancez un nouveau `shell` avec la commande `bash`. Une fois cette commande exécutée, faites afficher la valeur de `A`, le PID du `shell` courant et celui de son père. Que constatez vous ?

Question 3

Quittez le second `shell` ouvert dans la question précédente avec la commande **`exit`**. Et affichez la valeur de la variable `A` et le PID du `shell` courant. Que constatez vous ?

Question 4

Nous souhaitons maintenant lancer un deuxième `shell` à partir du `shell` courant qui héritera de la variable `A`. Exécutez les instructions nécessaires. Vous vérifierez que le deuxième `shell` a bien dans son environnement d'exécution une variable `A` initialisée à 30.

Question 5

Modifiez la valeur de la variable `A` dans le deuxième `shell`, quittez-le et affichez la valeur de la variable `A` du premier `shell`. Que constatez-vous ? Comment l'expliquez vous ?

Question 6

Ouvrez un terminal (que nous appellerons `terminal1`), initialisez à 30 une variable `A`, affichez la valeur de `A`, ouvrez un nouveau terminal à partir de `terminal1` en exécutant la commande `xterm` et affichez la valeur de `A` dans ce nouveau terminal. Que constatez-vous ?

Question 7

Dans `terminal1`, exportez la variable `A` et ouvrez un nouveau terminal (que nous appellerons `terminal2`) à partir de `terminal1`. Affichez la valeur de `A` dans `terminal2`. Que constatez-vous ? Que pouvez-vous en déduire ?

Question 8

Modifiez la valeur de la variable `A` dans `terminal2` et affichez la valeur de `A` dans `terminal1`. Que constatez-vous ? Comment l'expliquez-vous ?

Question 9

Modifiez la valeur de la variable `A` dans `terminal1` et affichez la valeur de `A` dans `terminal2`. Que constatez-vous ? Comment l'expliquez-vous ?

Question 10

Ouvrez un nouveau terminal depuis `terminal2` et affichez la valeur de `A` dans ce nouveau terminal. Que constatez-vous ? Comment l'expliquez-vous ?

Question 11

Ouvrez un terminal depuis les menus (de la même façon que vous avez ouvert `terminal1`) et affichez la valeur de `A` dans ce nouveau terminal. Que constatez-vous ? Que pouvez-vous en déduire ?

Exercice 25 – Redirection des entrées-sorties

Dans cet exercice, lorsqu'un processus doit transmettre une information à son fils, il le fera par l'intermédiaire d'un paramètre.

Soit le script de calcul suivant :

```
#!/bin/bash
# calcul.sh
if [ $# -ne 3 ]
then
    echo Il faut trois parametres
    exit 1
fi
if [ "$2" != "+" ] && [ "$2" != "-" ]
then
    echo "l'operateur doit etre + ou -"
    exit 2
fi
expr $1 $2 $3
```

Question 1

Écrivez un script `ensemble_calculs.sh` qui utilise le script `calcul.sh` pour exécuter plusieurs calculs. L'exécution

Prompt% `./ensemble_calculs.sh "1 + 2" "5 - 7" "8 - 1" "8 - 8"`

doit produire l'affichage suivant :

```
3
-2
7
0
```

Question 2

Nous souhaitons maintenant écrire une nouvelle version du script `ensemble_calculs.sh`. Les résultats des calculs doivent maintenant être écrits dans un fichier dont le nom sera passé en premier paramètre du script `ensemble_calcul.sh`. Si ce premier paramètre correspond à un répertoire existant, l'exécution doit s'arrêter. S'il correspond à un fichier existant, le contenu de ce dernier doit être remplacé par le résultat des calculs. Voici un exemple d'exécution.

```
Prompt% ls
ensemble_calculs.sh
Prompt% ./ensemble_calculs.sh fic_res "1 + 2" "5 - 7" "8 - 1" "8 - 8"
Prompt% cat fic_res
3
-2
7
0
```

Vous devez proposer deux versions du nouveau script `ensemble_calculs.sh` (un script `ensemble_calculs1.sh` et un script `ensemble_calculs2.sh`). Une des deux solutions ne doit provoquer qu'une seule ouverture du fichier. L'autre solution doit provoquer plusieurs ouvertures (une par écriture).

Question 3

Nous souhaitons maintenant modifier les scripts précédents pour que l'écriture des résultats positifs se fasse dans un fichier différent de celui des résultats négatifs ou nuls. Si un des deux fichiers est un sous-répertoire du répertoire courant, l'exécution doit s'arrêter. Si un des fichiers existe déjà, son contenu doit être remplacé par le résultat des calculs. Si `fic_pos` et `fic_neg` ne correspondent pas à un sous-répertoire du répertoire courant, vous devez obtenir l'exécution suivante :

```
Prompt% ls
ensemble_calculs.sh
Prompt% ./ensemble_calculs.sh fic_neg fic_pos "1 + 2" "5 - 7" "8 - 1" "8 - 8"
Prompt% cat fic_neg
-2
0
Prompt% cat fic_pos
3
7
```

Vous devez proposer deux solutions, une dans laquelle les instructions d'écriture dans les fichiers se trouvent dans le nouveau script `ensemble_calculs.sh` et l'autre dans laquelle les instructions d'écriture dans les fichiers se trouvent dans le nouveau script `calcul.sh`. Dans chacun des cas, vous créerez de nouveaux scripts de manière à ne pas écraser vos réponses aux questions précédentes.