

Exercices TP

Recopiez le contenu du répertoire /Infos/lmd/2011/licence/ue/li219-2012fev/TP5 qui contient tous les fichiers nécessaires pour la réalisation de ce TP et en particulier le fichier `compte_rendu_TP5.txt` que vous devez compléter au fur et à mesure et soumettre à la fin de la séance.

Exercice 40 – Constatation des commutations

Soit le script `affiche.sh` suivant :

```
#!/bin/bash
# affiche.sh
if [ $# -ne 1 ] ; then
    echo Il faut un parametre : un entier
    exit 1
fi
i=0
while [ $i -lt $1 ] ; do
    echo $$ : $i
    i=`expr $i + 1`
done
```

Question 1

Quel est l’affichage produit par l’exécution de la commande suivante ?

Prompt% `./affiche.sh 5`

Question 2

Quel est l’affichage produit par l’exécution de la commande suivante ?

Prompt% `./affiche.sh 5 & ./affiche.sh 5`

Exécutez la commande plusieurs fois (jusqu’à ce que vous obteniez un affichage ne correspondant pas à une alternance d’affichage entre les deux processus) et expliquez ce que vous obtenez. Qu’en concluez-vous ?

Question 3

Nous considérons maintenant le script `affiche2.sh` suivant :

```
#!/bin/bash
# affiche2.sh
if [ $# -ne 1 ] ; then
    echo Il faut un parametre : un entier
    exit 1
fi
i=0
while [ $i -lt $1 ] ; do
    echo -n $$ :
    echo $i
    i=`expr $i + 1`
done
```

Exécutez plusieurs fois la commande suivante (jusqu’à ce que les modifications faites sur le script aient une conséquence sur l’affichage). Qu’en concluez-vous ?

Prompt% `./affiche2.sh 5 & ./affiche2.sh 5`

Exercice 41 – Mise en évidence des incohérences provoquées par les commutations

Soit le script `ecriture.sh` suivant :

```
#!/bin/bash
# ecriture.sh
if [ $# -lt 1 ] ; then
    echo Il faut au moins un parametre
    exit 1
fi
for elem in "$@" ; do
    if [ ! -e "$elem" ] ; then
        echo premier $$ > "$elem"
    else
        echo suivant $$ >> "$elem"
    fi
done
```

Question 1

Exécutez deux fois de suite la commande suivante et expliquez le contenu des fichiers a, b et c obtenus.

Prompt% `./ecriture.sh a b c`

Question 2

Soit le script `lancement_ecriture.sh` suivant qui permet d'exécuter en concurrence deux processus `ecriture.sh`.

```
#!/bin/bash
# lancement_ecriture.sh
if [ -f f1 ] ; then
    rm f1
fi
if [ -f f2 ] ; then
    rm f2
fi
if [ -f f3 ] ; then
    rm f3
fi
./ecriture.sh f1 f2 f3 & ./ecriture.sh f1 f2 f3
```

Exécutez le script `lancement_ecriture.sh` jusqu'à ce que le contenu des fichiers `f1`, `f2` et `f3` obtenus soit différent de celui obtenu à la question précédente (la différence ne doit bien sûr pas se limiter à la valeur des PID). Expliquez le résultat obtenu et dites pourquoi il n'est pas satisfaisant.

Question 3

Modifiez le script `ecriture.sh` pour que l'exécution de la commande `lancement_ecriture.sh` donne obligatoirement un résultat non satisfaisant (vous devez forcer une commutation à un moment judicieusement choisi).

Question 4

Pour que l'exécution de la commande précédente ne puisse pas produire de résultat non satisfaisant, nous proposons la solution suivante dont le principe est qu'un processus qui souhaite accéder à un fichier bloque tous les autres processus tant que nécessaire. Voici les étapes à respecter :

- Avant d'accéder pour la première fois à un fichier, un processus doit envoyer un signal `SIGSTOP` aux autres processus qui souhaitent accéder à un fichier (quel qu'il soit), de manière à bloquer leur exécution.
- Pour que les identités de ces processus soient connues, un processus qui veut accéder à un fichier doit "s'inscrire", c'est-à-dire écrire son identité (PID) dans un fichier commun à tous les processus (`fic_PID`) et dédié aux inscriptions.

- Une fois que le processus a fini de travailler sur le fichier, il réveille les processus stoppés en leur envoyant un signal `SIGCONT`. De cette façon, tous les processus sauf un sont bloqués lors des accès aux fichiers.

Chaque processus doit effectuer son inscription avant toute autre instruction. De plus, vous devrez veiller à ce qu'un processus ne s'envoie par un signal à lui même.

Modifiez le script `ecriture.sh` et vérifiez que le résultat de l'exécution de la commande `lancement_ecriture.sh` est correct.

Question 5

Cette “solution” n'est pas un bon moyen de résoudre les problèmes mis en évidence dans cet exercice. Quels sont les défauts que vous identifiez ?