

ISS - Initiation aux Systèmes d'exploitation et au Shell

LU2IN020

TD 11 – L'autre synchronisation : la barrière

Julien Sopena

novembre 2021

Cette onzième semaine revient sur les mécanismes de synchronisation étudiés la semaine dernière. Ainsi, nous avons déjà vu comment ces derniers permettent à plusieurs processus de travailler sur une même ressource partagée. Dans cette séance, nous verrons comment ces mêmes mécanismes permettent de mettre en place des barrières, *i.e.*, d'attendre la fin d'une action ou d'un calcul fait par un autre processus.

Exercice 1 : Ping-pong, on passe au double

Dans cet exercice, on revient sur l'exercice du ping-pong du neuvième TD. Mais cette semaine vous, allez utiliser la commande de synchronisation `iss_synchro` introduite la semaine dernière.

Question 1

Proposez une implémentation des scripts `ping.sh` et `pong.sh` tel que le script `ping.sh` lance le processus `pong.sh` avant d'afficher **"ping"**. Les deux processus alterneront alors toutes les secondes les affichages : **"pong"**, puis **"ping"**, puis **"pong"**, ...

Question 2

Pourrait-on implémenter la même fonctionnalité avec un `wait` ?

Question 3

Est-il possible d'implémenter ces scripts en n'utilisant qu'un seul verrou ? Si oui comment ? Sinon pourquoi ?

Question 4

On veut implémenter un double impliquant 4 processus : deux pour l'affichage du **"ping"** et deux pour l'affichage du **"pong"**. Le principe reste inchangé : un des 2 processus `ping_double.sh` affiche **"ping"**, puis un des processus `pong_double.sh` affiche **"pong"**.

Proposez une implémentation basée sur la commande `iss_synchro` qui permette d'assurer l'alternance des affichages désirée. Le lancement se fera en exécutant la commande `./ping_double.sh`.

Question 5

Quelle modification faut-il faire dans vos scripts pour permettre un arrêt complet en faisant en **Ctrl+C**? Vous veillerez entre autres à détruire correctement les verrous créés.

Question 6

Votre implémentation donne-t-elle des garanties sur le nombre d'affichages fait par les différents processus?

Question 7

On veut maintenant coder un **ping-ping-pong**, avec deux processus `ping_simple_double.sh` et un processus `pong_simple_double.sh`. Ce dernier ne fera son affichage que lorsque chacun des deux processus `ping_simple_double.sh` aura affiché la chaîne **"ping"**. Donnez une implémentation de ces deux scripts, sachant qu'ils seront lancés par la commande `./ping_simple_double.sh`.

Question 8

Pour finir, proposez une nouvelle version des scripts `ping_simple_double.sh` et `pong_simple_double.sh` qui maintienne dans un fichier `compteur.txt` le nombre total d'affichages (**"ping"** ou **"pong"**).

Exercice 2 : Deux types de synchronisation

Question 1

Dans l'exercice précédent est-ce que tous les unlock sont faits par des processus ayant fait un lock sur le verrou?

Question 2

Ces deux dernières semaines étaient consacrées à la synchronisation des processus. Nous y avons utilisé la commande `iss_synchro` pour résoudre deux types de problèmes différents. Lesquels?

Question 3

Quelle est la différence entre une barrière et un `wait`?

Question 4

Dessinez deux chronogrammes illustrant ces deux types d'utilisations et les patterns de programmation associés. Pour simplifier, on prendra le cas d'un multi-cœurs, *i.e.*, le cas où les processus peuvent s'exécuter en parallèle.