

Programmation Système

TP n° 3 : synchronisation de processus

8-9 février 2016

L'utilisation de la fonction `popen()` n'est pas autorisée dans ce TP.

Exercice 1 : exclusion mutuelle

1. Écrire une commande « `lonely` » qui affiche `je suis seul` toutes les cinq secondes sur sa sortie standard. Pour garantir qu'au maximum une instance de « `lonely` » s'exécute à tout instant, le programme devra créer (de façon exclusive, `O_EXCL`) un fichier `/tmp/lonely.lock`, dans lequel il écrira son `pid`. Si une autre instance est déjà en cours d'exécution, le programme retourne 127 immédiatement. Au bout de `MAX` affichages, le programme termine, sans oublier de supprimer le fichier `/tmp/lonely.lock`.
2. Lancer, tuer, puis relancer votre programme. Quel est le problème ?
3. Modifier votre programme pour qu'il utilise un verrou exclusif de type BSD, en faisant toujours en sorte qu'il quitte immédiatement si une autre instance est en cours d'exécution.

Exercice 2 : aller-retour

Écrire un programme qui crée deux processus. Le père commencera par écrire le message "Salut, fiston !" dans un tube; le fils lira alors le message, répondra "Salut, papa !" dans un deuxième tube puis terminera; le père lira alors le message du fils avant de terminer à son tour. Pourquoi faut-il créer deux tubes ? Vérifier le comportement de votre programme à l'aide de la commande « `strace -f` » (sous Linux; « `dtruss -f` » sous Mac OS).

Exercice 3 : redirection

Écrire un programme qui affiche le nombre d'utilisateurs connectés à la machine sur laquelle on l'exécute à l'aide des commandes « `w` » et « `wc -l` »; ce programme ne devra créer aucun fichier sur le disque.

Exercice 4 : « `grep` » parallèle

Écrire un programme « `occurrences` » qui prend en paramètre une expression régulière r et un nombre arbitraire n de noms de fichiers, puis qui affiche le nombre total d'occurrences de r dans les n fichiers en exécutant n instances de « `grep -c` » en parallèle. Le père lira tous les résultats et calculera le nombre total d'occurrences.

Modifier votre programme pour qu'il exécute au plus 4 instances de « `grep -c` » en parallèle. Quelle version est la plus rapide ?

Exercice 5 : rotation de jeton

Dans cet exercice, n processus doivent écrire dans un même fichier. Pour éviter les conflits, ils sont organisés en *anneau* : le processus p_i est relié à p_{i+1} par un tube¹. Un jeton (entier) circule dans cet anneau, et seul le processus en possession du jeton est autorisé à accéder au fichier.

Écrire un programme qui crée 10 processus puis utilise ce procédé pour faire en sorte que chaque processus réalise exactement 1000 écritures (de son `pid` par exemple) dans le fichier.

Que se passe-t-il si l'un des processus meurt avant que tout soit terminé ?

1. les indices sont pris modulo n