

## Programmation Système

### TP n° 4 : verrous, tubes nommés

15-16 février 2016

#### Exercice 1 : verrou brisé

Tester les comportements respectifs des verrous POSIX et BSD <sup>1</sup>, après chaque étape, dans les contextes suivants <sup>2</sup> :

1. Écrire un programme qui pose un verrou sur un fichier `test_verrou`, duplique le descripteur, puis ferme le second descripteur.
2. Écrire un programme qui pose un verrou sur un fichier `test_verrou`, ouvre une seconde fois `test_verrou`, puis ferme la seconde ouverture.
3. Écrire un programme qui pose un verrou sur un fichier `test_verrou`, crée un processus fils, puis (dans le père) ferme le descripteur.

La page <http://0pointer.de/blog/projects/locking.html> dresse un constat désabusé des capacités des verrous sous Unix... mais les nouveaux verrous OFD devraient enfin proposer une solution acceptable.

#### Exercice 2 : gestion de logs

Le but de cet exercice est d'écrire un démon semblable au démon « `syslog` » tournant sur tout système Unix.

1. Écrire un programme « `log-daemon` » qui commence par créer un tube nommé `/tmp/logger` accessible, uniquement en écriture, par tout utilisateur. Votre programme ouvrira ensuite le tube en lecture, lira les données qui arrivent, et les affichera sur sa sortie standard. Lorsque la lecture indique un marqueur de fin de fichier, votre programme fermera le tube, l'ouvrira de nouveau, puis recommencera. Comme il s'agit d'une ébauche de démon, vous devrez être particulièrement soigneux dans la gestion des erreurs.  
Que se passe-t-il si vous lancez votre programme sans qu'aucun écrivain n'ait ouvert le tube ? Vérifiez votre hypothèse à l'aide de la commande « `strace` » <sup>1</sup>.
2. Écrire un programme « `logger` » qui formate ses paramètres de ligne de commande sur le tube `/tmp/logger` en les terminant par un caractère de fin de ligne. Testez votre programme. Que se passe-t-il si vous exécutez « `logger` » alors que « `log-daemon` » ne s'exécute pas ? Que se passe-t-il dans le démon quand votre écrivain meurt ? Testez vos hypothèses à l'aide de la commande « `strace` » <sup>1</sup>.
3. Modifier le programme « `log-daemon` » pour qu'il fasse précéder chaque ligne qu'il affiche de la date courante (vous pouvez vous servir des fonctions `localtime()` et `strftime()` avec le format `"%F %T"`).

1. et OFD si vous disposez d'une distribution assez récente

2. sous Linux, vous pouvez par exemple introduire des pauses de 5 secondes et consulter `/proc/locks`