

Série d'exercices 5 : Synchronisation

- 5.1 Vous voulez réaliser un certain traitement d'image en 3 étapes sous forme de pipeline. La première étape est le chargement de l'image, ensuite l'application d'un effet, et la dernière étape est la sauvegarde. Chaque item est passé au suivant par un tampon circulaire. Comment réaliser cette architecture avec des sémaphores? Décrivez quels sont les rôles de chaque fils d'exécution (producteur et consommateur).
- 5.2 Dans quel cas on voudrait utiliser un verrou asymétrique read/write plutôt qu'un verrou à exclusion mutuelle?
- 5.3 Plusieurs processus veulent ajouter de l'information à la fin d'un fichier avec `mmap()` simultanément. Décrivez comment vous pourriez faire pour éviter qu'un processus écrase l'information d'un autre processus en train d'écrire, tout en étant le plus efficace possible?
- 5.4 Linux supporte uniquement le verrouillage de fichier coopératif (contrairement à un verrouillage obligatoire), c'est-à-dire que si on verrouille un fichier avec `flock()`, il faut que l'autre processus utilise `flock()` lui aussi. En d'autres termes, on ne peut pas s'assurer d'avoir l'accès exclusif à un fichier en forçant un autre fichier à attendre. Quels sont les avantages et les inconvénients de cette situation?
- 5.5 Réalisez deux scripts pour expérimenter avec les verrous sur un fichier. Le premier script modifie un fichier périodiquement, tandis que l'autre doit le sauvegarder périodiquement. Utiliser la commande `flock` pour assurer l'exclusion mutuelle entre les deux script.
- 5.6 Faites les exercices du site <https://deadlockempire.github.io/>