



Outils pour l'enseignement des sémaphores

Contexte

Les **sémaphores** sont un des outils de base pour synchroniser l'exécution concurrente de processus ou de threads, et sont donc utilisés dans divers enseignements en informatique. Cet enseignement est en général assez délicat puisque les étudiant·e·s manipulant des sémaphores sont exposé·e·s à plusieurs difficultés à la fois.

- Maîtriser l'interface d'un sémaphore (opérations `Init(nb)`, `P()` et `V()`).
- Simuler (mentalement) l'exécution d'un programme concurrent (multithread ou multiprocessus).
- Concevoir des algorithmes concurrents.
- Si des implémentations de sémaphores sont utilisées en TP, maîtriser le modèle mémoire d'une application multithread ou multiprocessus, et connaître l'impact de l'ordonnanceur système.

Objectifs du projet

Le but principal de ce projet est de développer des outils pour aider à l'enseignement des sémaphores.

Un premier objectif sera d'évaluer les possibilités offertes par le Model Checker de **SimGrid** dans ce contexte, en écrivant des codes de synchronisation spécifiques à SimGrid. Cet outil émule l'exécution d'un code en parcourant tous les chemins d'exécution possible, en vérifiant que certaines propriétés sont toujours respectées à certains points de l'exécution du code émulé. Idéalement, SimGrid permettra soit de montrer que le code étudié respecte toujours les propriétés voulues, soit donnera un contre-exemple : un chemin d'exécution permis par le code où des propriétés ont été violées (par exemple, plusieurs processus sont en même temps dans la même section critique).

Un second objectif sera de visualiser les contre-exemples d'exécution au mieux, afin que les étudiant·e·s puissent analyser plus simplement pourquoi leur code est invalide.

Enfin, un troisième objectif sera de voir s'il est possible de faire une couche logicielle d'interception de sémaphores classiques (POSIX) pour les émuler de manière plus ou moins transparente dans SimGrid.