

[TD-5] Inversion de priorité

Dans ce TD, nous utiliserons les développements de **TD-3** (en particulier **Calibrator** et **CpuLoop**) et **TD-4** pour mettre en évidence le phénomène d'**inversion de priorité** abordé dans le cours Multitâches-1. Référez-vous au cours Multitâches-2 et la documentation de référence Posix pthread pour le paramétrages possibles d'un `pthread_mutex_t`.

Complétez la classe **Mutex** avec une option de protection contre l'inversion de priorité par héritage de priorité comme indiqué sur la **figure 1**.

Programmez le scénario proposé dans le cours Multitâches-1 et montrez par la mesure que la tâche A (la plus prioritaire) termine son exécution plus rapidement lorsqu'on utilise un mutex protégé contre l'inversion de priorité.

ATTENTION: Le processeur étant multicore, pour mettre en évidence l'inversion de priorité, il est nécessaire d'occuper tous les cores à 100% sauf un et faire tourner votre TD sur cet unique core laissé libre. Par exemple, sur un processeur quad-core, vous devez faire tourner 3 threads à haute priorité avec une boucle qui occupe le CPU à 100% (par exemple une boucle infinie qui incrémente un compteur) pour que les tâches du TD tournent sur l'unique core laissé libre.

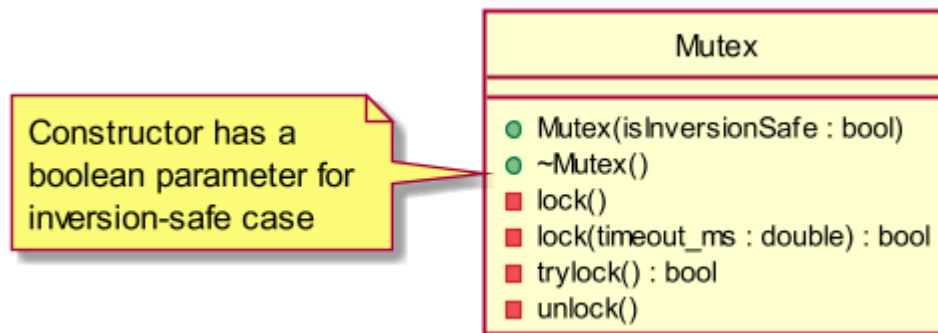


Figure 1 : Classe Mutex avec option de protection contre l'inversion de priorité.