

TD8

1 Verrous et sémaphores

On rappelle les primitives des verrous et sémaphores :

Verrous :

- `threading.Lock()`
- `Lock.acquire([blocking=1])`
- `Lock.release()`

Sémaphores :

- `class threading.Semaphore([value])`
- `acquire([blocking])` (P)
- `release()` (V)

Question 1 : Rappeler le fonctionnement de ces primitives et les différences entre un verrou et un sémaphore.

2 Mise en place

2.1 Lecteur / Écrivain

On dispose d'une ressource partagée et de deux types d'intervenants :

- les lecteurs qui veulent consulter la ressource ;
- les écrivains qui veulent la modifier.

Les lecteurs peuvent accéder de façon concurrente à la ressource mais les écrivains doivent le faire de manière exclusive.

Question 2 : Proposez une mise en œuvre à l'aide de sémaphores du problème des lecteurs/rédacteurs. Faites attention aux problèmes d'inter-blocage ou de famine.

2.2 Producteur / consommateur

On s'intéresse maintenant au problème du producteur / consommateur. On dispose de deux intervenants :

- un producteur qui produit des objets ;
- un consommateur qui les consomme.

Question 3 : On suppose que l'on dispose d'une seule case pour stocker un objet. Proposez une mise en œuvre à l'aide de sémaphores ou de verrous de ce problème en évitant inter-blocages et famines.

Question 4 : Faire le même exercice si on dispose maintenant d'un ensemble fini de case sous forme d'un tableau de n cases.

Question 5 : Pour finir, proposer une mise en œuvre dans le cas précédent si on dispose de plusieurs producteurs et consommateurs.