UFR IM²AG

UNIVERSITÉ Grenoble Alpes

DU ISN Seconde année, Système et réseaux

Synchronisation en Python

1 Producteurs- Consommateurs : algorithme de synchronisation

Le problème des producteurs consommateurs est un problème classique de synchronisation en informatique et peut être trouvé dans de nombreux contextes. Il s'agit d'une situation où plusieurs entités manipulent un tampon partagé, contenant plusieurs cases. Les producteurs produisent des valeurs qu'ils déposent dans le tampon, alors que les consommateurs consomment ces valeurs en les enlevant du tampon. Les producteurs bloquent quand le tampon est plein, alors que les consommateurs bloquent quand le tampon est vide.

Nous avons vu en cours l'utilisation des *conditions* dans la solution du problème du parking. Le but ici est d'utiliser les *conditions* afin de résoudre le problème des producteurs-consommateurs.

2 Synchronisation en Python : Le problème du parking

Nous avons vu la solution du problème du parking en cours. Le programme Python vous est fourni (`parking_conditions.py`). S'échauffer en faisant exécuter et en observant le programme.

3 Programmation des Producteurs- Consommateurs

En vous basant sur l'algorithme construit en section I et en utilisant les primitives Python de manipulation de *conditions*, programmer une simulation du problème des producteurs-consommateurs. Un squelette de programme vous est fourni (`prod-cons-A_COMPLETER.py`).

4 Lecteurs-Rédacteurs

Dans le problème des lecteurs / rédacteurs, un ensemble de lecteurs souhaitent accéder à une même ressource en lecture, tandis qu'un ensemble de rédacteurs souhaitent accéder à cette même ressource en écriture. Une solution correcte à ce problème doit garantir qu'un rédacteur n'accèdera jamais à la ressource commune en même temps qu'un lecteur ou qu'un autre rédacteur. En d'autres termes il peut y avoir un nombre illimité de lecteurs qui accèdent en concurrence à la ressource, mais un redacteur devra toujours y accéder seul.

Concevoir et programmer une solution de synchronisation pour ce problème.

5 le problème du parking avec des sémaphores

Etudier et comprendre la solution du parking utilisant des sémaphores (`parking_semaphores.py`).