

Rapport de projet : Producteurs/Consommateurs

I. Introduction:

Notre projet s'agit d'un exemple informatique de synchronisation de ressources dans différents contextes de programmation concurrente, notamment en environnement multi-thread. Il s'agit de partager entre deux tâches (producteur et consommateur) une zone de mémoire tampon utilisée comme une file.

Le producteur génère une donnée et la dépose dans la file, simultanément, le consommateur retire les données de la file. La coopération Producteurs-Consommateurs est une communication de messages entre des processus appelés Producteurs et d'autres appelés Consommateurs.

II. Choix faits:

Nous avons commencé par deux **class (Producer/Consumer)** où chacun contient les méthodes **wait()**, **run()** et **produce_item (Producer) / consume_item (Consumer)**.

- **produce_item()** : Elle ajoute un chiffre aléatoire (**nums**) dans un tableau (**items**) avec **append** ou dans une **Queue**.
- **consume_item()**: Elle retire le premier chiffre ajouté dans un tableau avec **pop()**, les chiffres aléatoires nous permettaient de voir si la valeur retirée était bien la première de la liste.
- **wait()** : **time.sleep()** y est intégré avec un temps aléatoire.
- **run()** : Le producteur va attendre puis produire, et le consommateur va attendre puis consommé.

III. Essai:

Nous avons essayé de faire des **Lock()** pour que le consommateur ne consomme pas un produit qui a été verrouillé mais cela n'a pas fonctionné, nous obtenons :

Le producteur qui produit sans arrêt sans que le consommateur puisse en consommer.