13/12/2016

Damien SIEST et Thibaud VEGREVILLE

POLYTECH GRENOBLE

Rapport

Projet producteur/consommateur



Objectif atteints et temps de réalisation :

Objectif 1 : 15h

Objectif 2 : 15 min

Objectif 3 : 6h

Explications & tests :

**Objectif 1 & 2 :**

Test 1 :

Dans ce premier test nous avons voulu vérifier qu'il pouvait y avoir autant de lecteur qu'on veut sur la ressource disponible. On choisit alors de mettre 10 consommateurs en paramètre et 0 producteurs.

On se rend compte que lorsqu'un lecteur demande l'accès il peut se mettre en attente si la ressource n’est pas disponible et on a alors un blocage.

Test 2 :

Cette fois nous réalisons le test avec un producteur et trois consommateurs pour voir les effets de blocage :

On constate que les producteurs ne peuvent pas bloquer les consommateurs et vice versa, dû au wait() des consommateurs.

Ces caractéristiques font parties des propriétés attendues du programme à la version v1.

Test 3 :

On effectue le test avec 10 producteurs uniquement pour observer le bon fonctionnement de ces derniers.

On remarque ici un blocage : il ne peut y avoir plusieurs producteurs sur une même ressource.

Avec ces 3 tests, on peut garantir que la version v1 du programme s'exécute en adéquation avec les consignes quel que soit le nombre de producteurs et de consommateurs.

**Objectif 3 :**

On peut rajouter des threads via l’exécution principale en appelant les méthodes de ProdCons qui notifient l’Observateur. Ce dernier possède des journaux d’évènements qui sont soit particulier, soit lié à un type de thread, comprenant aussi la notation en temps de l’évènement.

**Suite des objectifs :**

Suite au retard et problème diverse, nous n’avons pu nous attaquer à la suite du projet. Cependant, en reprenant le code des sémaphores, il aurait été facile de l’implémenter juste en modifiant la fonction put ou get de ProdCons et en ajoutant un attribut dans les messages.