



Projet systèmes concurrents/intergiciels

Rapport Provisoire de la version centralisé de Linda

Kamal Hammi
Mohamed Ikich
Salahdine Ouhammali
Groupe G10

Département Sciences du Numérique - Deuxième année
2021-2022

1 Réalisation de Centralized Linda

On a implanté la version centralisé du Linda à l'aide des moniteurs pour gérer la synchronisation des différents Threads qui l'utilisent, et en protégeant l'accès à l'espace des Tuples. Pour cela, on a utilisé une liste pour stocker les Tuples, un moniteur, deux HashMap de Tuple et condition pour les deux modes READ et TAKE, et en fin une HashMap de Tuple et HashMap des Callbacks avec leur eventMode. EventRegister est implanté en dernier, en cherchant directement dans l'espace des tuples, le tuple correspondant au motif demandé lorsque le eventTiming est FUTURE, sinon on le stocke dans la liste des CallBacks pour l'appeler par la suite, dans la méthode Write(), avec le tuple à ajouter à l'espace des Tuples, si il correspond bien au motif demandé.

2 Plan de Test Complet

2.1 Tests basiques des méthodes de centralizedLinda

On réalise des tests basiques avec des threads utilisant les différents méthodes (READ,TAKE,READALL,TAKEALL,TRYTAKE,TRYREAD) :

1. Test basique de la méthode READ.
2. Test basique de la méthode READALL.
3. Test basique de la méthode TAKEALL.
4. Test basique de la méthode TRYREAD.
5. Test basique de la méthode TRYTAKE.
6. Test de 2 threads avec la méthode READ.
7. Test de 2 threads avec la méthode TAKE.
8. Test de 2 threads avec la méthode TAKEALL.
9. Test de 2 threads avec la méthode READALL.
10. Test de 2 threads avec la méthode TRYTAKE.
11. Test de 2 threads avec la méthode TRYREAD.
12. Test de 2 threads avec les méthodes TAKEALL et READALL.

2.2 Tests de Call Back

1. 2 Threads utilisant le même Call Back avec les mêmes modes : READ et IMMEDIATE
2. 2 Threads utilisant le même Call Back avec les mêmes modes : TAKE et IMMEDIATE
3. 2 Threads utilisant le même Call Back avec les mêmes modes : READ et FUTURE

4. 2 Threads utilisant le même Call Back avec les mêmes modes : TAKE et FUTURE
5. 1 Thread utilisant un Call Back avec les modes : READ et IMMEDIATE.
Et un autre Thread utilisant un Call Back avec les modes : TAKE et IMMEDIATE.
6. 1 Thread utilisant un Call Back avec les modes : READ et IMMEDIATE.
Et un autre Thread utilisant un Call Back avec les modes : TAKE et FUTURE
7. 1 Thread utilisant un Call Back avec les modes : READ et FUTURE.
Et un autre Thread utilisant un Call Back avec les modes : TAKE et IMMEDIATE
8. 1 Thread utilisant un Call Back avec les modes : READ et FUTURE.
Et un autre Thread utilisant un Call Back avec les modes : TAKE et FUTURE
9. 1 Thread utilisant un Call Back avec les modes : TAKE et IMMEDIATE.
Et un autre Thread utilisant un Call Back avec les modes : TAKE et FUTURE.
10. 1 Thread utilisant un Call Back avec les modes : READ et IMMEDIATE.
Et un autre Thread utilisant un Call Back avec les modes : READ et FUTURE

2.3 Tests de Asynchronous Call Back

Les mêmes tests que Call Back normal, en changeant le type Call Back donné au EventRegister par Asynchronous Call Back.