INSA RENNES

Département informatique

TP SYSTEME AVANCE no 3
Synchronisation Avancée

Partie 1: événements

Dans ce TP, on se propose de réaliser la mise en oeuvre de la notion d'événement vue en cours. Comme nous ne souhaitons pas modifier le langage Java, nous allons créer des événements sous forme d'objets possédant deux méthodes (read et write). On se limitera ici a des événement à valeur entière. Pour cela on définit la classe suivante :

```
Class IntEvt {
      < attributs à définir >
      public IntEvt(){...}
      public int read(){...} // rend la valeur de l'événement ou bloque l'appelant
      public write(int i){...} // ecrit une valeur dans l'événement
}
```

On rappelle que :

- seul le Thread qui a créé un évènement est autorisé a écrire (exécuter write)
- un événement ne peut être écrit qu'une fois
- l'écriture libère tous les Threads en attente de lecture
- un Thread qui lit un événement non encore écrit se bloque

Question 1 : mettre en oeuvre la classe IntEvt
Question 2 : reprendre l'exemple du cours pour tester votre mise en oeuvre.

Partie 2: Introduction aux techniques dites « Balking » (state dependence)

Nous allons mettre en oeuvre un programme lecteur/rédacteur dans lequel lorsqu'un Thread se voit refuser l'accès à l'objet partager il effectue une autre action et donc NE SE BLOQUE PAS !

Pour cela, un Thread essaiera régulièrement de réaliser une action sur l'objet (ecriture ou lecture); lorsqu'il se verra refuser l'accès, il affichera "Le Thread x fait autrechose. Dans cet exercice il n'y a donc ni sémaphore, ni condition

Question 1 : réaliser la mise en oeuvre sans priorité. L'objet partagé contiendra des méthodes `lire` et `ecrire`. Il ne doit y avoir aucun blocage dans le programme. On utilisera des exception pour gérer les refus d'accès.
Question 2 : modifier la mise en oeuvre pour donner priorité aux rédacteurs. Il faut que l'objet partagé se souvienne qu'il y a des demandes d'écriture non satisfaites ... C'est un peu plus compliqué qu'avec les `wait/notify`

Ces notions seront revue en cours