Applications Mobiles - TD 2

‡ ‡ ‡

Thread Java : Syncronisation

Dans ce Td, nous nous intéressons à la synchronisation entre différents Thread.

I - Communication asynchrone

Exercice I.1

Communication asynchrone - V1

Reprendre la problématique de l'exercice 3 du TD précédent [1]. `TH1` resp.`TH2` sont périodiques de période *700 ms* resp. *300 ms* et s'exécutent durant *30s*. `TH1` doit déposer une donnée (par exemple une chaîne) dans un module de données. `TH2` affiche la valeur lue dans ce module de données.

1. Version 1 : utilisez un moniteur « classique » pour le partage de données entre ces deux Thread.
2. Version 2 : lecture multiple
   (a) Modifier votre code pour utiliser l'outil `Lock` ;
   (b) Ajouter 2 autres Thread de période *500 ms* et *1000 ms* réalisant une opération de lecture.

Exercice I.2

Communication asynchrone - V2

Reprendre le code de l'exercice précédent :

1. proposer une implémentation par sémaphores.

II - Communication synchrone

Exercice II.1

Boite aux lettres à *n* places sans écrasement - V1

Soient 4 thread :

- `TH1` et `TH2` déposent un message composé du nom du thread et du numéro d'instance du message dans une boite aux lettres à *n* places. Si celle-ci est pleine, `TH1` et `TH2` se retrouvent bloqués en attente d'une place libre.
- `TH3` et `TH4` retirent un message à la fois de la boite et affichent le contenu du message à l'écran. Si la boite est vide, les deux Thread se retrouvent bloqués en attente d'un nouveau message dans la boite.

1. Proposer une implémentation utilisant les interfaces `Condition`.
2. Faire plusieurs exécutions en faisant varier les périodes des différentes tâches afin de vérifier le bon fonctionnement des synchronisations de chaque côté de la boite.
3. Pour les plus rapides, proposer une implémentation générique de cette boite aux lettres. En effet, cet outil de communication entre Thread est très souvent utilisé.

Exercice II.2

Synchronisation par sémaphore

Soient deux Thread :

↬ `TH1` périodique [2], de période *500 ms*, qui après avoir affiché son nom provoque l'exécution de `TH2` [3]

↬ `TH2`, à réveil synchrone, affiche son nom à l'écran.

1. Proposez une implémentation utilisant un sémaphore de synchronisation.

1. I.e. version périodique classique
2. On utilisera ici une tâche périodique classique.
3. On implémentera « TH1 » en utilisant un `Thread` classique.

Exercice II.3

Boite aux lettres à n places sans écrasement - V2

Reprendre l'exercice II.1 :

1. Proposer une implémentation utilisant les sémaphores.

III - On mélange tout ? ? ?

Exercice III.1

Simulation d'un aéroport

On souhaite ici modéliser le fonctionnement d'un aéroport constitué d'une unique piste d'atterrissage et de décollage. Cette piste ne peut, évidement, accepter qu'un seul avion simultanément. L'aéroport dispose d'une tour de contrôle permettant de gérer l'accès à deux files d'attentes :

- *air* de taille N pour les avions souhaitant atterrir
- *sol* de taille M pour les avions souhaitant décoller.

Les avions sortent périodiquement du garage et demandent l'autorisation de décoller à la tour de contrôle en donnant leur nom. La tour de contrôle les place dans la file d'attente au sol.
De même, les avions en vol demandent, en donnant leur nom, à la tour de contrôle la possibilité d'atterrir. Cette dernière les place dans la file d'attente en l'air.
Une fois dans l'une ou l'autre des files d'attente, les avions doivent obtenir l'accès à l'unique piste afin d'atterrir ou de décoller.
Enfin, avant les premières demandes d'atterrissage ou de décollage, l'aéroport doit « s'initialiser ». Cette phase est réalisée par deux opérations s'exécutant en parallèle :

- préparation piste, qui prend 2,5 secondes
- Initialisation des systèmes de la tour de contrôle, qui prend 3,5 secondes

1. Analyser les différents processus nécessaires ainsi que les problèmes de communication/synchronisation, permettant de modéliser cet aéroport.
2. Implémenter votre solution permettant de simuler le fonctionnement de l'aéroport [4].

‡ ‡ ‡

4. Parmi tous les mécanismes vus en cours, vous mettrez en œuvre les plus simple permettant la résolution de ce problème