

05/10/2020

TP4 – Les threads en java et leur synchronisation

Objectifs : Le codage de serveurs nécessite l'utilisation des tâches concurrentes. Il est donc nécessaire de comprendre le fonctionnement du multitâche simple en java et mettre en évidence la nécessité de la synchronisation des threads lors de l'accès aux ressources partagées. La ressource partagée est ici la console `System.out` ; le TP met en évidence la problématique par le biais de l'affichage ; alors que dans une application serveur, il s'agira de protéger l'accès concurrent aux collections destinées à stocker les différentes données.

Prérequis : Récupérer le sujet du TP4 sur GIT. Installer le plugin Eclipse *ANSI Escape in Console*.

Travail à faire :

Comprendre le code source du TP4.

Le programme ajoute 10 tâches à raison d'une tâche par seconde.

Chaque tâche est identifiée (de 1 à 10).

Chaque tâche incrémente un compteur (tick) et affiche sur la console, à chaque incrément, une trace dans la colonne qui lui est réservée avec une couleur différente. Le démarrage, la fin ainsi que l'interruption de chaque tâche sont également tracés.

Chaque incrément dure 1 seconde sauf pour la tâche 4 (200 ms). Le tick 25 de la tâche 4 interrompt la tâche 0.

L'affichage des traces est artificiellement ralenti pour mettre en évidence la perturbation de l'affichage en cas de non synchronisation. En effet, la console `System.out` est une ressource partagée dont l'accès doit être protégé par un verrou.

```

loop 7
task 7 starts
task 7 tic 0
task 6 tic 2
[task 4 interrupts the task t0]
task 0 is interrupted
task 4 tic 25
task 5 tic 4
task 2 tic 10
task 3 tic 8
task 0 java.lang.InterruptedException: sleep interrupted
task 0 ends
task 1 tic 12
task 4 tic 26
task 4 tic 27
task 4 tic 28
task 7 tic 1
task 6 tic 3
task 4 tic 29
task 5 tic 5
task 3 tic 9
task 2 tic 11
task 4 tic 30
task 1 tic 13
task 4 tic 31
loop 8
task 8 starts
task 8 tic 0
task 7 tic 2

```

Figure 1: Affichage synchronisé

```

loop 7
task 7 starts
[ t a
pts the task t0]
task 0 is in t e rtrauspkte d6 k t 2
t a i s c
stjka asv4k a t.5li act n2g5.i
Icn t4e
rruptedException: sleep interrupted
task 0 ends
t a stka s3k t2i ct i8c
10 task 1 t i c 1 2t
ask 4 tic 26
task 4 tic 27
task 4 tic 28
task 4 tic 29 task 7 tic 1
task 5 tic 5 task 6 t i c
t a stka s3k t2i ct i9 c
task 4 tic 3 1 1
task 4 tic 3 0
task 1 tic 13
task 4 tic 31
task 4 tic 32
loop 8
task 8 starts
task 8 tic 0

```

Figure 2: Affichage non synchronisé

- Enlever le verrou et constater la perturbation de l’affichage lorsqu’il n’est pas synchronisé.
- Apprendre les principes de la programmation concurrente en java. Voir par exemple <http://rom.developpez.com/java-synchronisation/>
- Modifier le TP2 pour créer un serveur supportant les accès concurrents par de multiples clients simultanés.