
Java Lab - Threads

Exercice 1 : Threads, juste pour se chauffer...

Un "compteur" a un nom (Toto par exemple) et il compte de 1 à n (nombre entier positif quelconque). Il marque une pause aléatoire entre chaque nombre (de 0 à 5000 millisecondes par exemple).

Un compteur affiche chaque nombre (Toto affichera par exemple, "Toto : 3") et il affiche un message du type "*** Toto a fini de compter jusqu'à 10" quand il a fini.

Ecrivez la classe compteur et testez-la en lançant plusieurs compteurs qui comptent jusqu'à 10. Voyez celui qui a fini le plus vite.

Exercice 2 : Tri parallèle

Voici un algorithme de tri en ordre croissant d'une tranche de tableau comprise entre les éléments d'indices debut et fin :

```
trier(debut, fin) {
    si (fin - debut < 2) { // on a fini, pas d'appel récursif
        si (t[debut] > t[fin])
            echanger(t[i], t[j])
    }
    sinon {
        milieu = (i + j) / 2
        trier(debut, milieu)
        trier(milieu + 1, fin)
        triFusion(milieu) // tri fusion des 2 moitiés de la tranche du tableau
    }
}
```

On remarque que les 2 tris qui sont effectués avant la fusion sont indépendants l'un de l'autre et il est donc facile des les faire exécuter en parallèle par 2 threads.

[Voici](#) une version Java mono-tâche de cet algorithme. Vous pouvez coder votre propre version si vous le souhaitez.

En utilisant la méthode `join()`, codez une version multi-tâche de cet algorithme. Si vous lancez des threads depuis la méthode `main()`, n'oubliez pas d'attendre la fin de leur exécution avant d'afficher le résultat final (ceux pour lequel le tableau "trié" s'affiche tel qu'il était au début, pas trié, me comprendront !).

[Voici](#) un squelette de classe que vous pouvez utiliser.

Exercice 3 :

Modifiez le code de Trieur pour qu'il hérite de Runnable et pas de Thread. Testez.

Exercice 4 :

Codez une nouvelle version en utilisant cette fois-ci `wait()` - `notify()` au lieu de `join()`.

Relancez plusieurs fois (au moins une dizaine de fois) votre programme pour essayer de repérer des éventuels problèmes liés au multi-tâche.

Exercice 5 :

Calculer le produit matriciel en multi-tâche. Pour rappel:

Si $A = (a_{ij})$ est une matrice de type (m, n) et $B = (b_{ij})$ est une matrice de type (n, p) , alors leur *produit*, noté $AB = (c_{ij})$ est une matrice de type (m, p) donnée par :

$$\forall i, j : c_{ij} = \sum_{k=1}^n a_{ik} b_{kj} = a_{i1} b_{1j} + a_{i2} b_{2j} + \cdots + a_{in} b_{nj}$$
