

Programmation multi-coeurs — TD 1

Du parallélisme à la concurrence

Exercice 1. *Asynchronie.* Décrire toutes les sorties possibles pour le programme ci-dessous.

```
Thread t = new Thread(() -> {
    System.out.print("a");
    new Thread(() -> { System.out.print("b"); }).start();
    System.out.print("c");
});

System.out.print("d");
t.start();
System.out.print("e");
t.join();
System.out.print("f");
```

Exercice 2. *Parallélisation.* Ecrire un fonction `double[][] product(int n, double[][] M1, double[][] M2)` qui calcule le produit de deux matrices carrées M_1 et M_2 de taille $n \times n$. Le calcul devra être parallélisé entre $O(n^2)$ threads. Comparer la complexité en temps du programme parallèle et du programme séquentiel.

Exercice 3. *Le savant fou.* En visite pédagogique dans un laboratoire, vous et le reste de votre groupe êtes capturés par un savant fou. Il fait l'annonce suivante :

« J'ai mis en place une pièce qui contient une lumière et un interrupteur qui la contrôle. Actuellement, la lumière est éteinte. Vous pouvez discuter toute la soirée d'une stratégie, mais à partir de demain vous serez isolés dans vos cellules et ne pourrez plus communiquer. Chaque jour, je tirerai aléatoirement l'un d'entre vous qui passera la journée dans la pièce. Il pourra alors déclarer "Nous avons tous visité la pièce". S'il a raison, vous aurez la liberté. Sinon, JE VOUS MANGERAI TOUS ! »

- a. Quelles propriétés doivent être vérifiées par une solution au problème ?
- b. Proposez une stratégie gagnante (on peut supposer que la lumière est éteinte au début).