Programmation parallèle avec Pthreads

1. Objectifs

- Identifier les éléments de parallélisme dans un programme séquentiel donné, en vue de son exécution sur une architecture à mémoire partagée.
- Implémenter de manière efficace les fonctions de la bibliothèque **Pthreads** pour paralléliser un programme séquentiel, visant à une exécution efficace sur une architecture parallèle à mémoire partagée.
- Analyser et comparer les performances d'une exécution séquentielle et d'une exécution parallèle du même programme afin d'évaluer l'accélération obtenue.

2. Outils

- Compilateur **C** et API **Pthreads**.

3. Travail demandé

a. Le problème à traiter dans ce TP, sur lequel les trois objectifs énoncés précédemment devront être appliqués, n'est pas imposé ; il appartient aux étudiants de faire leur sélection. Par exemple, ils peuvent choisir de se pencher sur un problème lié à la cryptographie, notamment l'algorithme **AES** en mode **CTR**.

b. Il est essentiel de présenter le fonctionnement de l'algorithme qui illustre le problème sélectionné.

c. Le pseudo-code séquentiel du problème sélectionné doit être exposé de manière claire et structurée.

d. Il est important d'expliquer la pertinence de la parallélisation du problème sélectionné en s'appuyant sur les éléments de parallélisme identifiés dans son pseudo-code séquentiel.

e. La stratégie adoptée pour paralléliser le problème sélectionné doit être clairement expliquée, en précisant pourquoi certaines parties ont été choisies pour la parallélisation et le type de partitionnement retenu pour ces segments du problème.

f. La solution parallèle résultant de cette stratégie doit être représentée sous forme de schéma, illustrant les tâches, ainsi que les communications et synchronisations nécessaires.

g. Il est nécessaire d'implémenter la solution parallèle proposée en tirant parti des fonctionnalités de la bibliothèque **Pthreads**.

h. Il est essentiel de mesurer le temps d'exécution de la solution parallèle du problème sélectionné pour différents nombres de threads (par exemple, 2, 4, 6, 8 et 16) et de le comparer à celui de l'exécution séquentielle du même problème. Il est également recommandé d'inclure, en plus du nombre de threads, d'autres paramètres tels que la taille des données, afin d'améliorer la comparaison et la discussion des résultats.

4. Les recommandations et les livrables

- Ce TP est obligatoire, et les étudiants peuvent s'organiser en monômes ou en binômes pour le réaliser.
- Les codes implémentant les solutions séquentielle et parallèle du problème sélectionné doivent être remis.
- Un rapport incluant les tâches demandées (**b**, **c**, **d**, **e**, **f** et **h**) doit également être remis.
- La date limite de soumission des livrables via la plateforme Classroom est fixée au **04 novembre 2024** à **08h00**.
- La notation du TP se fondera sur les critères suivants :

  - L'originalité du problème sélectionné.

  - Les arguments justifiant la parallélisation du problème sélectionné.

  - La clarté de la stratégie adoptée pour paralléliser le problème sélectionné.

  - La clarté de la solution parallèle proposée.

  - La qualité de l'analyse des comparaisons de performance.

  - La qualité de la structure du rapport soumis.

  - La présence des étudiants à la séance de TP du **28 octobre 2024** est obligatoire pour que les livrables soient acceptés à la date limite de soumission.