Programmation multi-cœurs
Contrôle continu 20/02/2020

Durée : 1 heure. (+ 20 minutes de tiers-temps) Tous les documents sont autorisés. La compréhension du sujet constitue un élément d'appréciation. En conséquence, seules les questions jugées pertinentes par l'examinateur recevront une réponse. Toute réponse doit être justifiée : la notation tiendra particulièrement compte de la précision des réponses et de la qualité des arguments. Le barême est donné à titre indicatif.

Exercice 1. *Question de cours (5 points).* Qu'est-ce qu'une condition de progression ? Quels sont les avantages et les inconvénients des différentes conditions de progression ? Rédiger environ 10 lignes au total.

Exercice 2. *Crible d'Ératosthène (7 points).* Le code Java séquentiel du crible d'Ératosthène, affichant les nombres premiers inférieurs à `max`, est donné ci-dessous. Écrivez une nouvelle version parallélisée entre n threads, où n est un argument supplémentaire à ajouter à la méthode `crible`. Votre code devra respecter les contraintes suivantes :

- chaque valeur de i devra être traîtée une et une seule fois,
- les nombres premiers devront être affichés dans l'ordre.

Discutez de la complexité de l'algorithme parallélisé en fonction de `max` et de n.

```
class Eratostene {
  static void crible(int max) {
    boolean hasDiv[] = new boolean[max];
    for(int i = 2; i<max; i++) {
      if(!hasDiv[i]) {
        System.out.println(i);
        for(int j = 2*i; j<max; j+=i) {
          hasDiv[j] = true;
        }
      }
    }
  }
}
```

Exercice 3. *Le passeur de l'Erdre (8 points).* Le but de l'exercice est de modéliser, sous la forme d'un moniteur, le comportement de la ligne N3 de la Tan : le passeur de l'Erdre est un bac effectuant des aller-retour entre les arrêts Petit-Port et Port-Boyer, sur les deux rives de l'Erdre. Quand des passagers arrivent à un arrêt, ils doivent attendre l'arrivée du bac avant d'embarquer puis de repartir dans l'autre sens.

Écrire une classe Java (ou C++) contenant deux méthodes, `portBoyer()` et `petitPort()` pouvant être appelées respectivement par les voyageurs souhaitant aller de Port-Boyer à Petit-Port, et ceux souhaitant aller de Petit-Port à Port-Boyer (les deux méthodes étant symétriques, on pourra ne décrire que `portBoyer()` et expliquer comment obtenir `petitPort()` en commentaires). Le bac devra respecter les contraintes suivantes :

- Le bac ne pourra pas partir d'un arrêt s'il y a des voyageurs sur le quai (il n'y a pas de limite de capacité dans le bac).
- Le bac ne pourra pas partir d'un arrêt si des voyageurs du dernier trajet ne sont pas descendus.
- Le bac peut voyager à vide pour aller chercher des passagers sur l'autre quai, mais sans compromettre la première contrainte.
- À chacun de ses voyages entre les deux arrêts, un et un seul message devra être affiché dans la sortie standard, indiquant le sens de voyage et le nombre de voyageurs à bord.

Indice : utilisez quatre compteurs de voyageurs : un pour les voyageurs de chacun des quais, et un pour les voyageurs à bord du bac dans chaque sens.