

Licence 2 — Info4B

Examen du mardi 10 juin 2025

Durée 2h • Documents autorisés : une feuille A4 recto-verso

Le barème est donné à titre indicatif



Si un exercice vous conduit à faire des hypothèses, indiquez-les clairement sur votre copie.

Rédigez et justifiez précisément les réponses aux questions.

Exercice 1 - 6 pts

1. Comparer la notion de verrou et la notion de moniteur.
2. Qu'est-ce qu'un buffer ? Quelle est son utilité ?
3. Quels sont les rôles du noyau d'un système d'exploitation ?

Exercice 2 - 6 pts

On souhaite stoker, dans une hashtable en Java, le classement de joueurs de go dont un extrait est donné ci-après :

Dai Junfu 732

Hwang In-Seong 727

Dréan-Guénaïzia Benjamin 641

Chen Chongxiao 630

1. Pourquoi les hashtables sont des structures intéressantes du point de vue du système d'exploitation ?
2. Définir et instancier la hashtable pour y stocker le classement des joueurs sachant qu'on fera des recherches sur le nom afin d'obtenir leur valeur.
3. Donner les lignes de code pour ajouter les deux premiers joueurs et leur classement.
4. Donner les lignes de code pour enregistrer la hashtable dans un fichier.
5. Pourquoi le hashtable n'est pas la structure la plus adaptée si on souhaite obtenir un classement trié ? Quelle structure conviendrait mieux ? Donner un exemple de stockage de données dans cette structure.

Exercice 3 - 8 pts

On considère un buffer implémenté sous la forme d'une liste. On veut créer plusieurs threads qui déposent des objets dans le buffer. Les objets ont tous le même type, ils appartiennent à la classe `Coordonnee` qui a deux attributs de type float (latitude et longitude). Lorsque le buffer est plein, les threads qui déposent les objets sont en attente de libération de place. D'autres threads prélèvent des objets dans le buffer. Celui-ci est géré en FIFO (premier entré premier sorti). Si le buffer est vide, les threads clients attendent et sont informés si un nouvel objet est disponible.

1. Identifier les problèmes de concurrence éventuels. Les interactions peuvent-elles engendrer un verrou mortel ?
2. Quels sont les synchronisations nécessaires ?
3. Définir la classe `Buffer` et les méthodes pour ajouter des objets ou en retirer.
4. Définir la classe thread qui sert à déposer les valeurs dans un buffer.
5. Écrire le code Java pour instancier un objet `Buffer`, et 2 threads qui déposent des objets `Coordonnee` ainsi que 3 threads qui les prélèvent.
6. Comment arrêter les threads ?
7. Comment modifier le programme afin que les threads puissent communiquer avec le buffer au moyen du réseau.