

Licence 2 — Info4B

Examen du Jeudi 13 juin 2024

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. Expliquer le mécanisme de préemption. Comment est-il implanté dans le noyau du système d'exploitation ? À quoi sert-il ? Quel est son lien avec la notion d'interruption ?
2. Comment réaliser une attente passive sur une ressource avec les primitives de synchronisation de Java ? Donner l'extrait de code correspondant.
3. Quel est le lien entre la notion de port et de processus ?

Exercice 2 - 7 pts

On construit une table de hachage ayant 20 blocs pouvant contenir chacun un couple clé-valeur (la valeur étant dans notre cas une référence vers un objet). On souhaite réaliser un répertoire des joueurs d'échecs. La clé sera constituée de la concaténation du nom et du prénom du joueur. Les informations à stocker sont par exemple son niveau en ELO, sa date d'inscription, la fédération à laquelle il appartient, le nombre de parties qu'il a disputé, le nombre de parties nulles, le nombre de parties gagnées.

1. La fonction de hachage se sert de la somme des valeurs des caractères ASCII des deux premières lettres du nom. Définir complètement la fonction de hachage pour permettre de déterminer le bloc associé au nom.
2. Ranger les joueurs suivants dans une table de hachage de 20 blocs et noter le nombre de débordements. Liste des joueurs à insérer dans la table (prénom et nom) : Magnus Carlsen ; Garry Kasparov ; Fabiano Caruana ; Levon Aronian ; Wesley So ; Shakhriyar Mamedyarov ; Maxime Vachier-Lagrave ; Viswanathan Anand ; Vladimir Kramnik ; Veselin Topalov ; Hikaru Nakamura ; Ding Liren ; Aleksandr Grichtchouk ; Alireza Firouzja. Les valeurs des caractères ASCII sont données ci-dessous.

65→A 66→B 67→C 68→D 69→E 70→F 71→G 72→H 73→I 74→J 75→K 76→L 77→M
78→N 79→O 80→P 81→Q 82→R 83→S 84→T 85→U 86→V 87→W 88→X 89→Y 90→Z
91→[92→" 93→] 94→^ 95→' 96→` 97→a 98→b 99→c 100→d 101→e 102→f 103→g
104→h 105→i 106→j 107→k 108→l 109→m 110→n 111→o 112→p 113→q 114→r 115→s
116→t 117→u 118→v 119→w 120→x 121→y 122→z

TABLE 1 – Table ASCII des caractères

3. On utilise maintenant une table de hachage de 10 blocs pouvant contenir chacun deux couples clé-valeur. Adapter la fonction de hachage, ranger les données et compter le nombre de débordements. Proposer une stratégie pour diminuer le nombre de débordements.
4. On souhaite créer cette table de hachage dans un programme Java, et une deuxième table de hachage permettant de retrouver directement toutes les parties jouées par un joueur (une partie étant représentée par une chaîne de caractères comportant les différents coups). Donner le code Java correspondant, sachant que ces tables de hachage doivent pouvoir être sauvegardées dans des fichiers.

Exercice 3 - 7 pts

On considère n threads chacun ayant un identifiant (1 à n). Chaque thread k reçoit des messages du thread qui le précède ($k - 1$) effectue une opération sur le message et le transmet au thread suivant ($k + 1$). Le thread 1 crée les messages initiaux, le thread n les enregistre sur le disque.

1. Faire un schéma du système. On suppose que les threads s'échangent des entiers et que l'opération appliquée sur un message est juste une incrémentation. Donner le code des classes qui implémentent les threads avec les traitements particuliers pour le thread 1 et le thread n .
2. Donner le code de la méthode `main` pour instancier 10 threads et les lancer.
3. On modifie le système afin que chaque thread communique sur le réseau et envoie des messages sous la forme d'objets contenant des informations en plus de l'entier. Définir une classe message comportant un entier et une chaîne de caractère pour l'identifiant du thread.
4. Comment vont communiquer les threads? Donner les lignes à modifier/ajouter pour que la communication se fasse au travers du réseau.
5. Tous les threads ne fonctionnent pas au même rythme, on ajoute une file d'attente pour les messages reçus. Des problèmes de concurrences peuvent-ils se produire? Modifier le code pour implémenter une file d'attente de messages reçus et les traiter ensuite.