Licence 2 — Info4B

Examen du vendredi 10 Juin 2022

Durée 2h • Document autorisé 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. Quels sont les deux rôles essentiels du noyau d'un système d'exploitation ? Quels sont ses rôles plus spécialisés (en donner au moins 5) ?
2. Décrire les avantages et inconvénients des méthodes d'allocation de mémoire (contiguë ou non contiguë) pour la gestion des fichiers et pour la gestion de la mémoire RAM affectée aux processus.
3. Faire un schéma d'un ordonnanceur de type tourniquet à 4 files d'attentes avec des *quantums* de temps identiques, pour gérer un processeur multi-cœurs (par exemple 6 cœurs). Décrire précisément son fonctionnement.

Exercice 2 - 7 pts

On considère un système de gestion de stock qui doit traiter des messages de retrait ou d'ajout de produits dans le stock. Les messages sont des couples de la forme (code du produit, opération). Opération est un entier, positif s'il s'agit d'un ajout, négatif s'il s'agit d'un retrait.

1. Faire un schéma du système.
2. Les messages sont produits par plusieurs *threads* et mis dans une file d'attente. Écrire le code de la classe `FileMessages`, puis la méthode `main()` qui instancie la file, les deux *threads* et les démarre. Expliquer comment vous gérez la concurrence d'accès à la ressource file.
3. Deux autres *threads* consomment les messages de la file pour effectuer les modifications sur le stock qui est représenté par une *hashtable* :
   (a) Définir la *hashtable*, donner les lignes de code pour la déclarer et l'instancier. Quel est son avantage par rapport à d'autres structures de données ?
   (b) Écrire la méthode `run()` des *threads* qui lit les messages tant qu'il y en a dans la file et met à jour le stock de chaque produit contenu dans la *hashtable*.
   (c) Peut-il y avoir des problèmes de concurrence lors de l'accès à la *hashtable* ? Comment les résoudre ? Donner les modifications à faire sur votre code.

Exercice 3 - 7 pts

On souhaite concevoir un système de partage de calculs entre 3 machines `A`, `B`, `C`. La machine `A` crée un objet `o` de type `O` comportant une méthode `m` qui sera invoquée successivement sur chacune des machines. L'objet `o` est transmis par le réseau, il passe de `A` à `B`, puis de `B` à `C`, retourne en `A` et continue. Chaque machine à tour de rôle reçoit l'objet, invoque la méthode `m` et le renvoie à la suivante une fois la méthode terminée. Un compteur `timeToLive` dans l'objet indique quand cet enchaînement doit s'arrêter c'est-à-dire quand on cesse d'invoquer la méthode `m`. Le compteur est décrémenté à chaque passage (arrivée) sur une machine. Lorsqu'il passe à 0 la machine qui reçoit l'objet `o` le stocke dans un fichier.

1. Quelles abstractions du système d'exploitation disponibles dans le langage Java allez vous utiliser ?
2. Ce système fait-il apparaître des verrous mortels ou des problèmes de concurrence ?
3. Écrire le code de la classe `O`.
4. Écrire la portion de programme qui permet de recevoir un objet `o` du réseau, d'invoquer la méthode `m` puis de renvoyer l'objet `o` à la machine suivante.
5. Écrire la portion de code qui effectue le stockage de l'objet dans un fichier lorsque son exécution est terminée.