

## Licence 2 — Info42

Examen du Mardi 10 Mai 2016

Durée 2h • Documents autorisés

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 - 7 pts

Les réponses aux questions de cet exercice sont limitées à 1/2 page de texte par question (hors schémas et/ou extraits de code).

1. Expliquez la notion de sérialisation de Java (préciser ses avantages et inconvénients).
2. Définir et expliquer l'utilité de la notion de port en TCP/IP.
3. Un socket serveur, suite à l'exécution de la méthode `accept()` crée un socket de connexion et redevient disponible. Dans l'hypothèse où le socket résultant de l'acceptation est pris en charge par un thread, expliquer comment les paquets peuvent être attribués à ce thread alors que le port est de nouveau disponible ?
4. Un algorithme de type *first-fit* peut-il être utile dans le cas d'une stratégie d'allocation non contiguë de mémoire ?

### Exercice 2 - 7 pts

On considère un système qui enregistre les traces des interactions des utilisateurs d'une application Java. Cette application est déployée sur différents postes et communique par sockets. À chaque action sur l'interface, on enregistre : la date sous la forme d'un timestamp<sup>1</sup>, le nom de la fonctionnalité activée (par exemple enregistrer, rechercher, saisie adresse, etc.) et le nom de l'utilisateur qui effectue l'action. Par ailleurs, on sait quel utilisateur s'est connecté sur quelle machine avec le timestamp et le nom login de l'utilisateur. Les informations de connexion et les traces d'interactions sont stockées de manières séparées. Il n'y a pas deux événements pour le même timestamp.

1. Ranger dans une hashtable et dans un arbre binaire les données du tableau 1 en considérant le nom de l'utilisateur comme la clé. La fonction de hachage est la valeur en ASCII (ASCII(a)=97) du premier caractère du nom d'utilisateur, modulo 10 (il y a 10 zones comportant chacune 5 cases). Dans les deux cas, discutez des avantages, inconvénients des deux structures sachant sachant qu'on veut trouver rapidement toutes les connexions d'un utilisateur et toutes les connexions ayant eu lieu entre deux timestamps.
2. On considère maintenant les interactions. Pourquoi l'utilisation d'un arbre binaire basé sur le type de l'interaction (enregistrer, rechercher, etc.) n'est pas une solution satisfaisante ?
3. Toujours dans les cas des traces d'interactions, on utilise le timestamp comme clé.
  - (a) Ranger les éléments du tableau 2 dans un arbre binaire.
  - (b) Soit une hashtable ayant 10 blocs de 100 unités. La fonction de hachage opère sur les deux derniers chiffres du timestamp. Donner la définition complète de la fonction de hachage. Ranger les éléments du tableau 2 dans la hashtable.
4. On estime que sur 1 an il y a autour de 6000 connexions et 300 000 interactions.

---

1. C'est-à-dire le nombre de secondes écoulées depuis le 1er janvier 1970 00 :00 :00 UTC jusqu'à l'événement à dater

nom login	adresse machine	timestamp
berger	192.168.1.2	1462852181
zarou	192.168.1.4	1462852131
dupont	192.168.1.3	1462852101
durant	192.168.1.8	1462855181
admin	192.168.1.4	1462853011
pierre	192.168.1.2	1462856101
alx	192.168.1.2	1462859101
berger	192.168.1.2	1462862181
dupont	192.168.1.3	1462864101

TABLE 1 – Connexion des utilisateurs, nom, machine, timestamp

timestamp	type interaction	nom login
146285990	rechercher	alx
146286190	modifier adresse	alx
146286250	enregistrer	pierre
146286291	enregistrer	alx
146286454	enregistrer	pierre
146286460	rechercher	pierre
146287190	modifier adresse	alx

TABLE 2 – Trace des interactions

- (a) Donnez la profondeur de l'arbre binaire utilisant le timestamp comme clé? Que peut on dire de la profondeur de celui qui enregistre les connexions?
- (b) Quelle est l'influence du volume des données sur les tables de hachage, comment leur permettre de stocker au mieux ces données?

### Exercice 3 - 6 pts

On développe une application réseau pour effectuer des calculs sur des données hébergées sur différentes machines. Chaque machine envoie à la suivante le résultat de son calcul quelle a combiné avec ses données locales. Une machine maître collecte les résultats. Pour implanter ce système on utilise le port 8990 en TCP. Si une machine a plusieurs cœurs de processeurs, elle peut accepter plusieurs calculs. Dans les messages échangés entre les machines, figurent, en plus des données (une matrice de réels), la liste des machines qui doivent, successivement exécuter leur calcul et en fin de liste la machine maître.

1. Faire un schéma détaillé du système (on prendra le cas de 3 machines et un maître) en faisant figurer les parties serveur et client. Faire apparaître les adresses IP, les ports (vous pouvez utiliser des ports supplémentaires), les ressources. Déterminer les problèmes de concurrence éventuels. Comment les résoudre?
2. Quels sont les mécanismes de Java que vous allez utiliser pour implanter ce système?
3. Écrire l'extrait code de la classe principale de chaque machine pour : créer le socket serveur et créer un thread lors d'une connexion.
4. Définir les attributs de la classe Java relative aux messages (elle peut être composée).
5. Écrire l'extrait de code pour envoyer les données d'une machine à une autre (en général et non pas dans le cas de trois machines).