

Applications Réseaux : TD 4

Communication Client-Server : Threads et NIO

L3 Informatique – 2020/2021

A Threads Java

1. Comment utiliser les Threads en Java ? Quelle est la méthode principale pour un thread ?
2. Écrire une classe `SocketHandler` qui implémente `Runnable`. Le constructeur de cette classe prend un identifiant socket (classe `Socket`) comme paramètre et la méthode principale doit lire les données de cette socket, ligne par ligne, et les stocker dans un ensemble de `String`.

B Serveur Multi-Thread

1. Comment peut-on construire un serveur (multi-thread) qui lit en parallèle de plusieurs client TCP sockets ? Utiliser la classe `SocketHandler` du dernier exercice.
2. Si on crée deux objets de type `SocketHandler` avec le même identifiant de socket, que se passe-t-il ? Si les deux threads lisent de la même socket client et écrivent sur l'écran chaque ligne de données reçus, qu'allons nous obtenir comme affichage ?
3. Rappelons que les threads d'une même classe partagent leur mémoire. Si deux threads écrivent sur la même structure de donnée que passe-t-il ? Donner un exemple.
4. Supposons que le système d'exploitation impose une limite de 1000 threads par processus. Pour le serveur multi-thread que vous avez écrit, y a-t-il un moyen d'accepter la 1001e connexion d'un client ?

C Communication avec Java NIO (New I/O)

1. Comment utiliser `java.nio.channels` pour la communication entre une socket serveur et une socket client ?
2. Quels sont les avantages à utiliser Java NIO pour la communication entre sockets ?
3. Pourquoi utilise-t-on un `Selector` pour les communications asynchrones ?
4. Écrire un serveur Mono-Thread, c'est-à-dire un serveur qui utilise un seul thread pour gérer les connexions avec jusqu'à 10 clients. Comment le serveur peut-il savoir quelle socket client est prête à lire ?