

TP n°7

Sockets en mode connecté

Le but de ce TP est de manipuler les sockets en mode connecté en *Java*. Nous allons réaliser différentes applications distribuées qui pourront être exécutées localement ou testées sur plusieurs postes.



Avant de commencer le TP, n'oubliez pas de tester les différents programmes d'exemple des fiches concernant les sockets en mode connecté.

1 Prise en main de TCP - Shifumi

Nous souhaitons écrire une application distribuée constituée de deux applications. La première, nommée *serveur*, écoute sur un numéro de port donné (spécifié dans un fichier de configuration) et se met en attente de réception de messages (un entier compris entre 1 et 3). Dès qu'un message est reçu, il est affiché à l'écran. La deuxième application, nommée *client*, envoie un entier compris entre 1 et 3 au serveur, l'entier étant lu au clavier. L'adresse IP et le numéro de port du serveur sont spécifiés dans un fichier de configuration.

1. Écrivez les applications et vérifiez leur fonctionnement.
2. Cette application distribuée fonctionne-t-elle sur le modèle de communication client/serveur ?
3. Que se passe-t-il si le serveur n'est pas démarré et qu'un client envoie un message ?

La valeur 1 correspond à Pierre, la valeur 2 à Feuille et la valeur 3 à Ciseaux.

4. Modifiez le serveur pour qu'il réponde maintenant avec une valeur aléatoire puis une chaîne de caractères contenant un message (gagné, perdu ou égalité, par exemple).

Cette fois-ci, nous souhaitons envoyer des messages au format JSON. Le serveur retourne un document JSON contenant un code d'état (OK ou erreur), ainsi que la valeur générée (entre 1 et 3), ainsi que le message (gagné, perdu ou égalité).

5. Proposez un format pour les messages.
6. Modifiez vos applications.
7. Quelle est la différence entre un envoi d'une chaîne de caractères correspondant à un document JSON et un envoi d'un objet `JSONObject` sérialisé ? À votre avis, quel est le meilleur choix ?

2 Un peu plus loin avec TCP

Nous souhaitons modifier maintenant le client pour qu'il fonctionne indéfiniment jusqu'à ce que l'utilisateur décide d'arrêter (en saisissant une valeur autre que 1, 2 ou 3, par exemple). Le serveur tourne également indéfiniment pour permettre l'accès à plusieurs clients, les uns après les autres.

Questions

1. Modifiez le client et le serveur. Vérifiez leur bon fonctionnement.
2. Que se passe-t-il si plusieurs clients sont exécutés en même temps ?
3. Nous souhaitons que les utilisateurs puissent jouer en parallèle. Modifiez le code du serveur.
4. Modifiez le serveur pour qu'il affiche l'adresse IP et le numéro de port de chaque client qui se connecte.



Vérifiez bien que l'adresse IP et le numéro de port affichés soient bien ceux des clients et non ceux du serveur. . .