

Algorithmique parallèle et distribuée

Sockets TCP

Adressage en Java

Exercice 1

Manipulons un peu le protocole IP :

1. en utilisant la méthode statique *getByName()* de la classe *InetAddress*, cherchez à obtenir l'adresse IP de *www.univ-paris5.fr*;
2. obtenir l'IP à partir d'un nom d'hôte ne signifie pas que l'IP est accessible. Utilisez la méthode *isReachable(int timeout)* pour savoir si un nom d'hôte est accessible ;
3. Cherchez à obtenir l'IP que votre machine utilise sur internet (de la classe *InetAddress*). Cherchez à utiliser la méthode *getLocalAddress()* de la classe *Socket*. Vous utiliserez pour cela une socket non connectée (*new Socket()*). Que se passe-t-il ?

Un premier programme client serveur

Exercice 2

Reprenez l'exemple du cours. Transformez le de façon à ce que le serveur renvoie au client toute chaîne que ce dernier lui envoie, telle que chaque caractère soit converti en majuscule (utilisez la fonction *toUpperCase()*).

Améliorons notre programme

Exercice 3

Notre premier programme est maintenant fonctionnel mais un peu limité en fonctionnalités. Améliorons le un peu :

1. Transformer l'exemple précédent de manière à ce que le client puisse répéter le traitement jusqu'à l'émission d'un message de terminaison (la chaîne "STOP" par exemple) ;
2. Transformer le serveur de façon à ce qu'il puisse servir plusieurs clients simultanément.

Lançons nous sur le réseau local

Exercice 4

Nous disposons d'un réseau local, interagissons avec les autres machines :

1. Transformer votre serveur de façon à ce qu'il affiche la chaîne de caractère reçue, l'IP de l'émetteur de cette chaîne, puis qu'il émette un message de retour personnalisé à votre guise. Lancez votre serveur autorisant les connexions multiples ;
2. Lancez votre client, essayer d'interagir avec les serveurs lancés sur les machines voisines (possédant un serveur fonctionnel).