

TP Sockets

L'objectif de ce TP est l'initiation à la programmation répartie. Il consiste à programmer une application typique comportant de la répartition à l'aide des Sockets. Vous devez programmer les parties "client" et "serveur" de cette application à l'aide du langage JAVA et la faire exécuter dans un environnement fait de plusieurs machines virtuelles. Ces machines pouvant être sur une seule machine physique (plusieurs JVM locales) ou sur plusieurs machines distantes.

Question 1 :

On vous demande d'établir une communication entre un processus client et un processus serveur qui échangent des messages sous forme de lignes de texte. Le scénario est le suivant :

- ✓ Le programme serveur ouvre un socket d'écoute et attend une connexion.
- ✓ Le programme client reçoit en paramètre l'adresse IP et le port du serveur. Il crée un socket pour se connecter sur la machine/port du serveur.
- ✓ Le processus client commence par émettre un message et le serveur lui répond par un écho de cette ligne en majuscule.
- ✓ Au bout de 10 échanges, le client envoie un message de terminaison 'END' et ferme la connexion.

Question 2 :

On vous demande de créer deux programmes client et serveur simples permettant d'établir les services suivants :

- ✓ Le serveur reçoit une chaîne de caractère du client et doit déterminer si cette chaîne est un palindrome ou pas. Le serveur doit envoyer le résultat au client qui va l'afficher à l'écran.
- ✓ Le serveur reçoit une chaîne de caractères du client et doit calculer la fréquence de la première lettre dans cette chaîne. Le serveur doit envoyer le résultat au client qui doit l'afficher à l'écran.

Question 3 :

On vous demande de créer deux programmes client et serveur simples permettant d'établir les services suivants d'une calculatrice :

1. Le premier service : **calcul du factoriel**

Le serveur reçoit un entier positif du client.

Le serveur doit calculer le factoriel de cet entier et doit le retourner au client qui va l'afficher à l'écran.

2. Le deuxième service : **puissance**

Le serveur reçoit deux entiers différents du client.

Le serveur doit calculer la puissance du premier entier par le deuxième **par récurrence**.

Le serveur doit envoyer le résultat de la puissance au client qui doit l'afficher à l'écran.

3. Le troisième service : **conversion binaire en décimal**

Le serveur reçoit un nombre binaire du client.

Le serveur doit convertir ce nombre binaire en décimal.

Le serveur doit envoyer le résultat de la conversion au client qui doit l'afficher à l'écran.

4. Le quatrième service : **calcul de la fréquence**

Le serveur reçoit un nombre supérieur à 10000 du client (par exemple : 1542112).

Le serveur doit calculer la fréquence de chaque chiffre de ce nombre.

Le serveur doit envoyer le résultat au client qui doit l'afficher à l'écran.