

# Travaux pratiques N°5

## Gestion des Threads avec Java

Programmation Orientée Objet en Java

2<sup>ème</sup> Année Génie Informatique

Enseignant : Tarik BOUDAA

AU : 2018/2019

### Exercice 1 : QCM

Répondre au QCM extrait des examens de certification Java. (Ce QCM vous sera distribué en classe sous format papier).

### Exercice 2 : Produit matricielle en multitâches

Soit  $A(a_{ij})$  une matrice  $(m,n)$  et  $B(b_{ij})$  une matrice  $(n,p)$ , le produit de  $AB = (c_{ij})$  tel que :

$$\forall i, j : c_{ij} = \sum_{k=1}^n a_{ik} b_{kj} = a_{i1} b_{1j} + a_{i2} b_{2j} + \dots + a_{in} b_{nj}$$

On remarque que les calculs des  $c_{ij}$  sont indépendants les uns des autres, ainsi on peut donc effectuer ces calculs en parallèles. Ecrire un programme Java qui permet de faire la multiplication de deux matrices en calculant en parallèles les  $c_{ij}$  en utilisant plusieurs threads.

### Exercice 3 : Programmation réseau et Threads

L'objectif de cet exercice est l'écriture d'un système de chat multi-utilisateurs qui se présente sous forme d'un salon de discussion *chat room* et qui fonctionne sur un réseau local. Ce système de chat est composé de deux programmes : **Programme Client** et **Programme Serveur**.

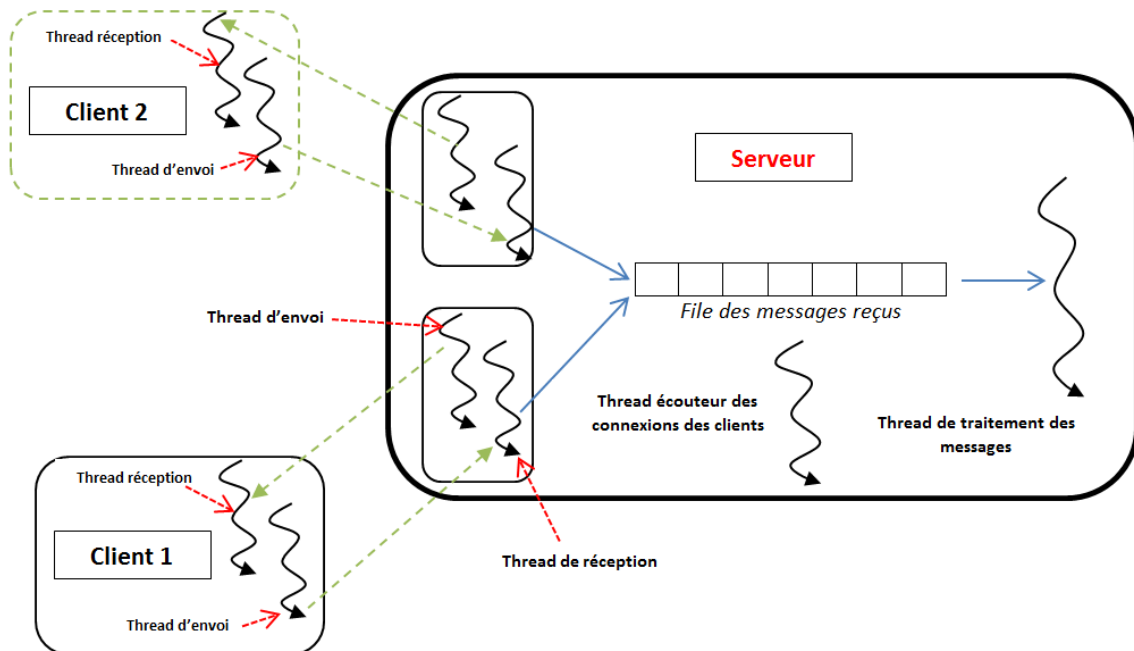


Figure 1

Le programme client permet d'envoyer les messages au serveur qui s'occupe de les dispatcher sur tous les utilisateurs connectés au serveur. Les opérations d'envoi et de réception des messages au niveau du client ne

doivent pas bloquer l'exécution de ses autres fonctionnalités, ainsi, il faut confier ces tâches à deux threads séparés (thread pour l'envoi et un autre pour la réception des messages). Le client permet également d'envoyer des fichiers sans bloquer les autres tâches en cours d'exécution.

Le serveur permet de centraliser la communication entre les nœuds qui utilisent le programme Client. Il s'occupe de gérer les connexions des clients et leurs messages, à chaque fois que le serveur reçoit un message d'un client il l'ajoute à une file d'attente des messages reçus, un autre thread s'occupe de dispatcher les messages de cette file sur tous les utilisateurs connectés sur le serveur.

La figure 1 donne une vue générale sur la conception de cette application, vous êtes donc invité à analyser et compléter cette conception puis l'implémenter en Java.