UNIVERSITE SAINT-JOSEPH
FACULTE D'INGENIERIE
ESIB

TRAVAUX DIRIGES ENVIRONNEMENTS DISTRIBUÉS
Séance 3 – Durée 01h30

Ce TD vous permettra de vous familiariser avec :

1- La personnalisation de la couche Transport RMI
2- Le Distributed Garbage Collector.
3- Les applications RMI qui mettent en œuvre des listes d'objets.

Partie I- Personnalisation de la couche transport

1. Vous devez commencer par vous assurer que le service RMI sur lequel vous allez faire la personnalisation s'exécute bien dans un seul folder. Un folder contenant le service et les classes nécessaires vous est fourni sur moodle. Compilez les classes qui vous sont fournies, démarrez trois terminaux puis testez votre service, en faisant attention à la configuration nécessaire à la ligne de commande. Expliquez.

2. Vous devez par la suite lire et comprendre comment s'est faite la personnalisation.

3. Quel est le design pattern utilisé pour l'implémentation des XOR Input et OutputStreams?

4. Au niveau de quelle instruction s'est fait le choix de nos Sockets et non des Sockets TCP par défaut?

5. Vous devez modifier la classe HelloImpl pour que l'on ait une classe serveur à part.

Partie II- Le Distributed Garbage Collector

1. Au niveau de cet exercice de même vous avez les classes nécessaires qui vous permettront de jouer avec votre DGC. Télécharez le folder correspondant sur moodle, compilez et exécutez.
2. Essayez de comprendre ce que le programme est en train de faire (cote client et cote serveur)

3. Pour jouer, suivez les étapes suivantes :

a. Start a new DOS console and go to the directory that contains the java code. Execute the following command: java RMIServer

b. Run two instances of the client.
Start two additional DOS consoles and go to the directory for the Java code. In each DOS console, prepare to execute the following command: java RMIClient but do not hit the enter key, just yet.
When both consoles are ready, hit the Enter key for both so that start running at roughly the same time.

Notice the output from the different consoles. The Server will show when it creates each object.
When each MessageObject is unreferenced, it will print a message and when the local garbage collector is about to destroy each MessageObject, they will print a message.

Notice that the local garbage collector does not destroy any objects until near the end of the run. You will also note that after the demonstration has finished it will take 10 minutes before the server complete releasing all of the objects.
This shows that the lease term of the remote objects has been set to 10 minutes. Since the clients have terminated, the server will wait until the end of the lease term before it marks the objects as being 'clean'.

4. Nous allons faire quelques configurations pour accentuer l'effet du Garbage Collecting :
   a. Au niveau de la propriété : java.rmi.dgc.leaseValue
   b. Au niveau de la taille maximale de la Heap au niveau de la JVM : java –Xmx1m RMIServer (This will run the Java Virtual Machine with a heap that can't grow: maximum 1Mb)

   4.1 Quelle est la valeur minimale permise de la Heap pour pouvoir démarrer votre application serveur?
   4.2 Fixez cette valeur à partir de la ligne de commande, en jouant aussi avec la valeur de la leaseValue du DGC.
   4.3 Exécutez plusieurs fois, par ex:
   java -Djava.rmi.dgc.leaseValue=600 –Xmx3m RMIServer
   Commentez. A quel moment se fait la finalization, et le dereferencement dans chacun des cas?

5. Faites les modifications necessaires au niveau de l'objet MessageObjet pour qu'il ne soit plus de type Remote, mais de type Serializable (MessageObjectSer) Qu'est-ce qui changera au niveau de l'affichage. Expliquez.

6. Essayez de retourner un objet dérivé de MessageObjectSer. Testez avec le client et le serveur dans 2 folders différents pour simuler qu'ils sont sur des machines differentes et expliquez.

Partie III- Mise en œuvre d'un Simple Banking System

1. Au niveau de cet exercice aussi vous avez les classes relatives à la mise en place d'un Simple Banking System avec Java RMI. Télécharez le folder correspondant sur moodle, **faites les corrections nécessaires au niveau du code**, compilez et exécutez.
2. Essayez de comprendre l'implémentation qui vous est proposée.
3. Est-ce que tous les objets Remote sont des objets enreigstrés au niveau du RMI registry ?
4. Est-ce que tous les objets Remote sont dérivés de UnicastRemoteObject ?