Client-serveur 2020-2021

pfister@connecthive.com

05/10/2020

TP5 – Un serveur multitâche

Objectifs : Le codage de serveurs nécessite l'utilisation des tâches concurrentes. L'objectif de ce TP est de passer du serveur monotâche du TP2 à une version multitâche et de vérifier qu'il peut répondre à plusieurs clients simultanément même s'ils maintiennent leur session.
Le second objectif est d'apprendre à utiliser les primitives de synchronisation, *synchronized, wait, notify*. Voir aussi par exemple http://rom.developpez.com/java-synchronisation/
Prérequis : TP2, TP3 et TP4. Compréhension des threads java.
Travail à faire :
Récupérer le squelette du TP5 ainsi que le corrigé du TP3 sur GIT.
Comprendre le code source du TP3 et vérifier son fonctionnement côté client et côté serveur.
En utilisant le squelette du TP5, transformer le serveur monotâche en serveur multitâche. Tracer chaque nouvelle connexion ainsi que les déconnexions.
Ajouter 2 mots au vocabulaire du serveur : *add* [element] et *remove*. Le mot *add* ajoute une chaîne dans une liste partagée dont la capacité est limitée. Si la capacité est atteinte la tâche (et la requête) est bloquée. Si une autre tâche retire un élément (mot *remove*), une des tâches en attente est notifiée ainsi peut achever son ajout. Le mot *remove* enlève le dernier élément de la liste. Si la liste est vide la tâche (et la requête) est bloquée. Si le retrait a réussi le serveur retourne au client l'élément retiré sinon il retourne un message d'erreur. Si le retrait a échoué sur un blocage, une autre tâche qui ajoute un élément provoque une notification à une des tâches bloquées qui sera débloquée et pourra achever le retrait.
De plus, l'accès à la ressource partagée est synchronisé.
Les tests se font avec le client du TP3.

```
	public static void addElement(ClientTask task,String elm) throws
InterruptedException {
		synchronized (sharedResource) {
			if (sharedResource.size() >= MAX_ELEMENTS) {
				log(task,"full, wait for room for at least one element");
				sharedResource.wait();
			} else {
				sharedResource.add(elm);
				log(task,"element added " + elm);
				sharedResource.notify();
			}
		}
	}
	public static String removeElement(ClientTask task) throws InterruptedException {
		String result = null;
		synchronized (sharedResource) {
			if (sharedResource.size() == 0) {
				log(task,"empty, wait for at least one element");
				sharedResource.wait();
			} else {
				result = sharedResource.remove(sharedResource.size() - 1);
				log(task,"removed element " + result);
				sharedResource.notify();
			}
		}
		return result;
	}
```

Expérimenter :

```
[serveur multisession] démarré sur :192.168.1.95:8051
en attente d'une connexion
en attente d'une connexion
task 0 starts
client 0
le client /127.0.0.1:59653 s'est connecté
le client demande: add 1
element added 1
en attente d'une connexion
        task 1 starts
        client 1
        le client /127.0.0.1:59656 s'est connecté
        le client demande: add 2
        element added 2
        le client demande: add 3
        element added 3
en attente d'une connexion
                task 2 starts
                client 2
                le client /127.0.0.1:59659 s'est connecté
                le client demande: add 4
                element added 4
en attente d'une connexion
                        task 3 starts
                        client 3
                        le client /127.0.0.1:59662 s'est connecté
                        le client demande: add 6
                        element added 6
en attente d'une connexion
                                task 4 starts
                                client 4
                                le client /127.0.0.1:59665 s'est connecté
                                le client demande: add 7
                                full, wait for room fo at least one element
en attente d'une connexion
                                        task 5 starts
                                        client 5
                                        le client /127.0.0.1:59668 s'est connecté
                                        le client demande: remove
                                        removed element 6
                                le client demande: remove
                                removed element 4
                                        le client demande: add 8
                                        element added 8
                                        le client demande: add 9
                                        element added 9
                                        le client demande: add 10
                                        full, wait for room fo at least one element
                                le client demande: remove
                                removed element 9
                                        le client demande: remove
                                        removed element 8
        le client demande: remove
        removed element 3
en attente d'une connexion
                                                task 6 starts
                                                client 6
                                                le client /127.0.0.1:59736 s'est connecté
                                                le client demande: remove
                                                removed element 2
                                                le client demande: remove
                                                removed element 1
                                                le client demande: remove
                                                empty, wait for at least one element
                                        le client demande: add 11
                                        element added 11
                                                le client demande: remove
                                                removed element 11
```