

TP : RMI

1) Il est difficile de broadcaster un serveur RMI car il faut envoyer tous les objets à tous les clients.

2) Le polling permet d'interroger une série de client pour leur demander lesquels d'entre eux sont intéressés par un service, ce qui permet de n'envoyer que les informations essentielles aux principaux concernés.

3) Cela peut poser un problème dans le cas où la chaîne de caractères viendrait à être supprimée car on perdrait l'intégralité de la conversation.

4)

.

. Implémentation du code

.

9) Le problème de la première version est qu'elle nécessite un thread du côté client qui demande constamment au serveur s'il y a un nouveau message. Une façon de faire plus propre aurait été de faire en sorte que le serveur prévienne le client lorsqu'il est concerné par un nouveau message. Pour cela, on peut utiliser un système de callbacks. Une fonction callback est une fonction qui est passée en argument à une autre fonction. Cette dernière peut alors faire usage de cette fonction de rappel comme de n'importe quelle autre fonction, alors qu'elle ne la connaît pas par avance.

Explication du code :

Cf. commentaires dans le code

Je crée une ArrayList qui va s'incrémenter des messages envoyés par les utilisateurs. La méthode getMsg permet de récupérer le dernier message envoyé, en prenant la taille du message en question en entrée afin de savoir à partir d'où dans l'ArrayList il faut envoyer. La méthode ecrireMsg permet de récupérer l'entrée clavier de l'utilisateur et de l'ajouter à l'ArrayList.