TP 5 : Multi-clients : Select

TP 5: Multi-Clients : select

Le but de ce TP est de voir comment gérer plusieurs clients simultanément sans parallélisme, ou presque. Le CR est à rendre sur AMETICE avant la prochaine séance.

Technique select

Appels bloquants vs non-bloquants

En rendant les sockets non-bloquantes, il est possible de gérer plusieurs clients avec un seul thread. Cette technique est appelée *select*, du nom de l'appel correspond en C.

Un serveur select

A l'aide des éléments du cours et du TD, écrire un serveur multi-client qui n'utilise qu'un seul fil d'exécution (Utiliser la classe

`Selector`
en Java.NIO)

Test de performance 1

Tester les performances de votre serveur pour un nombre de clients n:

- 1
- 2
- 10
- 100
- 1 000
- 5 000
- 10 000
- 50 000

qu'on a vu dans le dernier TP.

Serveur multi-connexion au delà de echo

Le but de cet exercice est de voir comment l'état d'une connexion peut être géré avec l'approche *select* (sans faire appel à de multiples threads).

Modifier un de vos serveurs monoclient Echo TCP de la sorte qu'il compte des lignes reçues et que chaque fois qu'une ligne est réçue, il renvoye le nombre total de lignes reçues sur cette connexion. Evidemment, vous utilisés une variable/attribut «nombre» pour compter.

Puis rendre ce serveur multi-client avec l'approche *select*. Il faudrait alors un compteur par connexion. Utilisez les méthodes attach et attachment de SelectionKey pour les gérer.

Valider l'implementation avec trois terminaux (le serveur et deux clients).

Modifié le: lundi 31 janvier 2022, 16:26