

TP/TD 12 : Un serveur de broadcast II

Le but de ce TP est de réaliser un serveur de *broadcast* : si un client envoie un message, ce message est automatiquement envoyé à tous les autres clients.

Question 1

Écrivez un serveur qui écoute sur un port fixé, attend qu'un client se connecte, ferme tout de suite la connexion et termine.

Utilisez `nc` pour tester: `nc localhost <port>`. Qu'observez-vous?

Question 2

Créez une structure de liste chaînée pour stocker vos clients, et acceptez en boucle toutes les nouvelles connexions – sans les fermer cette fois-ci.

Vérifiez avec `nc` que votre serveur autorise plusieurs connexions simultanées.

Question 3

Écrivez une fonction qui envoie un message (une chaîne de caractère se terminant par un retour à la ligne) à tous les clients actuellement connectés.

Question 4

En utilisant `select`, attendez en parallèle sur l'entrée standard et l'arrivée de nouveau client. Si l'utilisateur tape un message sur `stdin`, envoyez-le à tous les clients connectés.

Question 5

Ajoutez dans votre appel à `select`, tous les clients actuellement connectés. Si l'un d'eux a envoyé quelque chose, lisez le message et envoyez-le à tous les clients connectés.

Faites attention à bien gérer la déconnexion d'un client (`read` renvoie 0).

Question 6

Gérez le signal `SIGINT` : à sa réception, fermez toutes les connexions avec un client et quittez le serveur.

Question 7

Gérez les lignes partielles: si l'entrée d'un client arrive en deux fois (par exemple avec `echo -n foo; sleep 2; echo bar | nc localhost <port>` ou si plusieurs entrées arrivent en même temps. (Pour cela, il faut que chaque client ait son buffer, qui est vidé à chaque réception d'une nouvelle ligne.)

Question 8

Quand l'utilisateur entre `ping` sur l'entrée standard du serveur, envoyez un message "ping" à tous les clients. Déconnectez tous ceux qui n'ont pas répondu `pong` après trois secondes.