

TD5 : Sockets

Exercice 1 : Mise en œuvre du cours.

Reprendre les éléments serveurs et client du cours n°5 (sockets) pour écrire et tester deux programmes :

- Serveur.c : programme qui attend une connexion TCP sur le port 6000. Lorsque la connexion est établie, le serveur affiche **Client connecté**. Le serveur se met en attente de réception sur la socket. Il reçoit le message du client et l'affiche. Le serveur envoie alors un message au client (par la socket). Le serveur ferme sa socket de communication, puis ferme la socket d'écoute.
- Client.c : programme qui connecte le serveur sur le port 6000. Il affichera **Connexion OK**. Le client enverra un message au serveur, puis se mettra en attente de message depuis le serveur. Lorsque celui-ci arrive il l'affiche et ferme sa socket.

On pourra pour cet exercice utiliser l'adresse de serveur **127.0.0.1** et lancer les deux programmes sur la même machine, mais dans **deux terminaux différents**.

Exercice 2 : Discussion.

Modifiez le programme précédent pour faire une discussion à deux : chaque côté à son tour saisit et envoie un message (3 fois de suite). Ou une discussion à l'infini avec possibilité d'interrompre le dialogue (détection d'un caractère particulier au clavier).

Pour cet exercice on pourra faire l'essai avec un voisin dans le même réseau IP (Faire d'abord des essais de connectivité à l'aide de ifconfig et ping).

Exercice 3 : Serveur permettant des connexions successives.

Modifier le serveur de l'exercice précédent pour qu'il accepte successivement plusieurs connexions de la part de clients. On fera afficher au serveur l'adresse IP du client qui se connecte.

Exercice 4 : Serveur de temps.

Faire deux programmes client et serveur, permettant au client de se connecter au serveur pour obtenir l'heure du serveur. C'est-à-dire qu'un client se connecte et le serveur lui envoie sa date et son heure, puis se déconnecte. Il pourra accepter successivement plusieurs connexions.

Le client se contente d'afficher la date et l'heure qu'il reçoit.

Exercice 5 : Serveur permettant des connexions simultanées.

Modifier le serveur de l'exercice 3 pour qu'il accepte simultanément plusieurs connexions de la part de clients. On fera simplement afficher au serveur l'adresse IP du client qui se connecte et les messages que le client lui envoie. On utilisera un principe de forking pour gérer les différents clients.