

TP Sockets (2 x 3h) → rapport papier à rendre

Il est impératif de venir avec vos notes de cours.

Le rapport doit contenir l'ensemble du code de chaque question, présenté sous forme d'extraits commentés et expliqués. Il doit contenir des copies d'écran prouvant le bon fonctionnement de chaque partie.

Comme vu en cours, n'oubliez de gérer les erreurs (perror), ainsi que d'utiliser les outils unix à disposition, netstat...

Exercice 1 : mode non connecté, domaine AF_INET

Ecrire deux programmes, un client et un serveur :

Le client reçoit en argument sur la ligne de commande :

- Le nom de la machine hébergeant le serveur,
- Le port de ce serveur,
- Une chaîne de caractères,

Le serveur reçoit en argument sur la ligne de commande :

- Le port sur lequel il va attendre les connexions des clients

Objectifs :

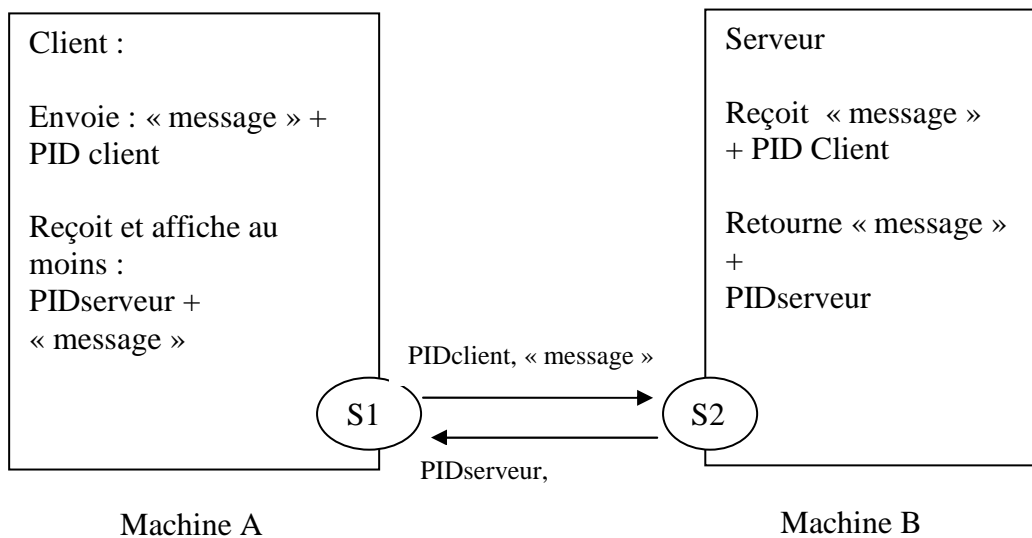
Le client envoie vers le serveur :

- Son numéro de PID
- La chaîne de caractères qui lui a été passée en argument

Le serveur :

- reçoit ces informations,
- les affiche,
- retourne au client son numéro de PID ainsi que la chaîne reçue.

Le client affiche le numéro de PID reçue du serveur et la chaîne de caractères



Communications en mode connecté

Exercice 2

Ecrire un serveur qui attend une requête de connexion sur son port d'écoute et qui affiche ensuite le flot de caractères émit par le client qui s'y connecte. Pour cela, le serveur reçoit en argument sur la ligne de commande le port sur lequel il va attendre les connexions des clients

A l'aide de votre navigateur préféré, connectez-vous sur ce serveur et observez les informations transmises. En déduire le codage d'une requête http (essayez plusieurs requêtes avec des demandes de pages web différentes) ?

Attention : pensez à changer les paramètres réseaux de votre navigateur pour qu'il n'utilise pas les proxy ! Notez aussi qu'il y'a deux saut de ligne après une requête...

Exercice 3

Ecrire un client qui reçoit en argument :

- Le nom du serveur auquel on va chercher à se connecter,
- Le numéro de port de ce serveur,
- Un nom de fichier.

Ce client doit se connecter à un serveur web standard (par exemple intra.esiee.fr) pour lui envoyer une requête http demandant l'affichage du nom de fichier donné en argument (par exemple index.html). Pour construire la requête, vous utiliserez les informations obtenues par le serveur de l'exercice 2.

Après avoir envoyé sa requête, votre client doit attendre et afficher les données envoyées par le serveur.

Vous vous limiterez au format de pages html pour ce TP

Attention : dans les salles TP de l'ESIEE vous ne pouvez accéder à un serveur extérieur puisque les machines sont derrière un proxy, seul www.esiee.fr et quelques machines internes sont accessibles directement, pour le reste du monde il faut passer par le proxy (voir exercice 5).

Exercice 4

Ecrire un petit serveur web qui réponde aux requêtes d'un navigateur (cad qu'il renvoie le fichier demandé). Ce serveur stockera dans un fichier local (log_file) toutes les requêtes c'est-à-dire l'adresse de la machine du client, la date et le nom du fichier demandé.

Ce serveur écoutera sur 2 ports (donnés en argument sur la ligne de commande). Il répondra normalement aux requêtes http sur le premier port, et retournera son fichier de log si un client s'y connecte par le second port.

Exercice 5 :

Faire en sorte que le client de l'exercice 3 puisse traverser le proxy, pour cela appliquez la même méthode que l'exercice 2 pour déterminer le protocole.