

Projet1 : sockets

Maitrise d'informatique/IUP2 GMI

5 octobre 2001

1 Client/serveur UDP et TCP

- Réaliser un client/serveur UDP (socket de type SOCK_DGRAM). Le client envoie des chaînes de caractères que le serveur affiche sur son terminal de contrôle.
- Adapter les programmes précédents pour permettre le transfert de fichiers du serveur vers le client.
- Réaliser un client/serveur proposant en TCP les mêmes services (envoi de chaînes de caractères, puis envoi de fichiers). On vérifiera notamment, pour les chaînes de caractères, que le nombre de connexions est limité par le paramètre de la fonction listen. Vérifier les sockets créées avec la commande netstat (port, processus attaché, état TCP, etc.).
Dans le cadre du transfert de fichier, intégrer une solution permettant au client de s'assurer que le fichier est intégralement transmis.

2 Dialogue avec un serveur existant

- Ecrire un client qui se connecte à un serveur HTTP pour récupérer une page web.
- Ecrire un client qui se connecte à un serveur SMTP pour envoyer un mail.

Les numéros de port associés à ces services se trouvent dans `/etc/services`. Les protocoles de dialogue entre un client et un serveur sont à trouver sur le web. A titre indicatif, le protocole HTTP est décrit dans le RFC (Request For Comments) 1945 ou 2068 et le protocole SMTP dans le RFC 821 (section 3.1). Pour HTTP, utiliser uniquement la commande GET.

La commande `telnet m p` établit une connexion TCP sur le port `p` de la machine `m` permettant alors de tester les protocoles.

3 Réalisation d'un talk multi-utilisateur

On souhaite réaliser un système client/serveur permettant de communiquer à plusieurs simultanément sur le réseau. L'implantation se fera en TCP.

Coté client :

Le client envoie des messages (texte) au serveur. Les messages sont envoyés ligne par ligne. Le client accepte les commandes suivantes saisies au clavier :

- `/connect <surnom> <machine> <port>` : se connecte au serveur et se déclare avec le surnom fourni en paramètre.
- `/quit` : quitte le client et le signale au serveur.
- `/who` : demande au serveur la liste des utilisateurs connectés.

Coté serveur :

Le serveur rediffuse tous les messages (texte) qu'il reçoit d'un de ces clients vers tous les autres clients connectés. Le serveur accepte les commandes suivantes saisies au clavier :

- `/who` : idem que `/who` coté client.
- `/kill <surnom>` : coupe la connexion du client correspondant au `<surnom>` et en informe les clients restants.
- `/shutdown` : pour arrêter le serveur.

Le serveur informe tous les clients de l'arrivée ou du départ d'un client.

Indication : c'est le retour-chariot qui va servir à séparer les différents messages.