

STI 3^{ème} année – Réseau

TP noté : IRC

J. Briffaut

1 TCP

Vous allez créer un serveur ECHO, tout message envoyé au serveur sera re-envoyé au client.

1.1 Server

Vous allez créer un serveur TCP écoutant sur le port 1234 et l'adresse localhost. Le serveur devra attendre un message, l'afficher, puis le renvoyer se remettre en attente d'un message.

- Exercice 1** Créez un fichier **tcpserver.c**
- Exercice 2** Ajouter une fonction **stop** habituelle pour gérer correctement les erreurs d'appels système
- Exercice 3** Créez une socket (**socket()**) de type TCP IPv4
- Exercice 4** Créez une structure d'adresse (**sockadd_in**) pour l'adresse **127.0.0.1** et le port **1234**
- Exercice 5** Associez votre socket a cette adresse (**bind()**)
- Exercice 6** Ecoutez au maximum 5 clients (**listen()**)
- Exercice 7** Acceptez un client (**accept()**)
- Exercice 8** Ajouter une boucle de 1000 itérations qui
 - attend la réception d'un message (**recv**)
 - affiche le message reçu
 - renvoi le message (**send**)
- Exercice 9** Tester votre serveur avec **echo -n coucou | nc -4 -w0 localhost 1234**
- Exercice 10** Que ce passe t'il ? Comment corriger ce problème ?

1.2 Client

Vous allez créer un client TCP envoyant des messages sur le port 1234 et l'adresse localhost. Le client enverra ECHO, puis devra attendre un message, l'afficher, et se remettre a envoyer ECHO, ceci 1000 fois.

- Exercice 11** Créez un fichier **tcpclient.c**
- Exercice 12** Ajouter une fonction **stop** habituelle pour gérer correctement les erreurs d'appels système
- Exercice 13** Créez une socket (**socket()**) de type TCP IPv4

Exercice 14 Créez une structure d'adresse (**sockadd_in**) pour l'adresse correspondant au serveur **127.0.0.1** et le port **1234**

Exercice 15 Ajouter une boucle de 1000 itérations qui

- envoie le message "ECHO" au serveur (**send**)
- attend la réception d'un message (**recv**)
- affiche le message reçu

Exercice 16 Tester votre client avec votre serveur (vous pouvez aussi vérifier avec Wireshark)

2 Serveur IRC

Exercice 17 Ecrire un serveur, écoutant sur le port 8000, supportant plusieurs clients connectés. Lorsqu'un client envoie un message, celui-ci est envoyé à tous les autres clients connectés.

3 Client IRC

Exercice 18 Ecrire un programme C qui permet de se connecter à votre serveur de chat, ce client devra permettre de :

1. Prendre en arguments l'adresse IP et le numéro de port de votre serveur
2. Demander un pseudo au lancement, qui sera envoyé au serveur, et le serveur l'utilisera pour prefixer tous les messages du client. Attention, les pseudos doivent être unique
3. Afficher l'heure de réception des messages (votre client devra récupérer l'heure via une requête NTP)

4 Fonctionnalités

Voici une liste des fonctionnalités à ajouter à votre client/serveur. Les parties sont indépendantes

Exercice 19 Gestion du pseudo (nickname)

1. Les pseudos doivent être uniques
2. Envoyer le message (commande) `/nickname nouveaupseudo` permet de changer de pseudo
3. La commande `/register pseudo mot-de-passe` permet d'enregistrer le pseudo, s'il n'est déjà pas enregistré par un autre utilisateur
4. La commande `/nickname pseudo mot-de-passe` permet d'utiliser un pseudo enregistré, et de forcer un utilisateur non-enregistré qui utilise ce pseudo à changer de pseudo
5. La commande `/unregister pseudo mot-de-passe` permet de supprimer le pseudo

Exercice 20 Commandes diverses

1. La commande `/exit` permet de quitter le serveur

2. La commande `/date` permet d'afficher l'heure serveur

Exercice 21 Message privé

1. La commande `/mp pseudo message` permet d'envoyer un message uniquement à l'utilisateur `pseudo`. Le message affiché sera préfixé par `private message from : pseudo-source`.

Exercice 22 Alerte

1. La commande `/alerte message` permet de notifier tous les utilisateurs avec un message important (en faisant bipper le PC, ou en affichant le message en gras/rouge)
2. La commande `/alerte pseudo message` permet de notifier un utilisateur avec un message important (en faisant bipper le PC, ou en affichant le message en gras/rouge)

Exercice 23 Envoi de fichier

1. La commande `/send pseudo fichier.txt` permet d'envoyer le fichier.txt à l'utilisateur `pseudo`. Le fichier s'affichera à l'écran, ou (mieux) sera enregistré dans le répertoire courant.

Exercice 24 En plus

1. Gérer la colorisation des messages (`/red /blue /green ...`)
2. Gérer des salons (pour les plus courageux) (`/join channel`)
3. Modérateur : l'utilisateur qui a créé un salon est modérateur, il peut expulser quelqu'un avec `/kick pseudo`