Buisson Valentin

Rapport Projet Network

1. Description

Le but de ce projet est de développer un serveur et un client en C. Le serveur doit pouvoir accepter la connexion de plusieurs lient en UDP ou en TCP. Une fois le client connecté, celui-ci doit s’identifier (mot de passe et nom d’utilisateur). Ces informations doivent être stocker dans une base de données SQLite3. Après identification, le client doit pouvoir accéder à la liste des fichiers sur le serveur, télécharger/uploader des fichiers, accéder à la liste de tout les clients connectés au serveur, envoyer des messages publics ou priver.

1. Fonctionnalités développées
2. Serveur :

Après avoir lancé le serveur, l’utilisateur a 3 possibilités : démarrer le serveur en TCP, démarrer le serveur en UDP, ouvrir le terminal.

a.1) Terminal : Le terminale permet à l’utilisateur d’accéder à la base de données. Une fois le terminal ouvert, l’utilisateur peut ajouter ou supprimer un utilisateur. Le terminal permet également d’afficher tous les utilisateurs dans la base de données. Il est également possible d’afficher tous les logs.

a.2) TCP/UDP : le mode UDP ou TCP permet de lancer le serveur et d’écouter les connections des clients. La seule différence entre ces 2 mode est la manière dont sont transféré les messages. Pour le reste, les fonctionnalités sont implémentées de la même manière.

-Identification :

Une fois connecté au serveur, le client doit se connecter pour avoir accès aux fonctionnalités, tant qu’il n’est pas connecté, il n’y a pas accès. Le serveur reçoit donc le nom d’utilisateur et le mot de passe du client, et vérifie dans la base de données s’il trouve une correspondance. Après avoir vérifié, le serveur renvoie -1 si aucune correspondance n’est trouvée ou 1 s’il en trouve une. Le serveur stock les sockets dans un tableau, et l’index du tableau de la socket est stocké dans la base de données. Ainsi pour trouver la socket d’un utilisateur précis, il suffit de rechercher son nom dans la base de données.

-Liste des utilisateurs :

Pour obtenir la liste des utilisateurs connecté, il suffit de faire une requête à la base de données. En effet, tous les utilisateurs connectés on l’index du tableau des sockets supérieur ou égale à 0. A l’initialisation du serveur ou a la déconnexion d’un client, cette valeur est remise à -1.

-Envoie d’un message public :

Lorsqu’un client souhaite envoyer un message à tous les utilisateurs, il envoie le message au serveur, et en utilisant la même méthode qu’avant, le serveur cherche dans la base de données tous les index supérieurs à -1, puis récupère la socket dans le tableau et lui envoie un message.

-Envoie d’un message privé :

Le principe du message privé est à peu près similaire au message public, le client envoie le message à transmettre au serveur, puis il envoie le nom de l’utilisateur à qui envoyer le message. Le serveur va donc ensuite chercher dans la base de données, l’index du tableau correspondant au nom d’utilisateur reçu. Il récupère ensuite la socket dans le tableau et lui envoie le message.

-Liste des fichier sur le serveur :

Lorsqu’un fichier est enregistré sur le serveur, il est mis dans le répertoire courant. Pour obtenir la liste des fichier disponible il suffit de lister tout les fichier dans le répertoire courant et d’envoyer cette liste à l’utilisateur.

-Uploader un fichier :

Envoie d’un message a une personne non connecté

Reset index to -1 in database