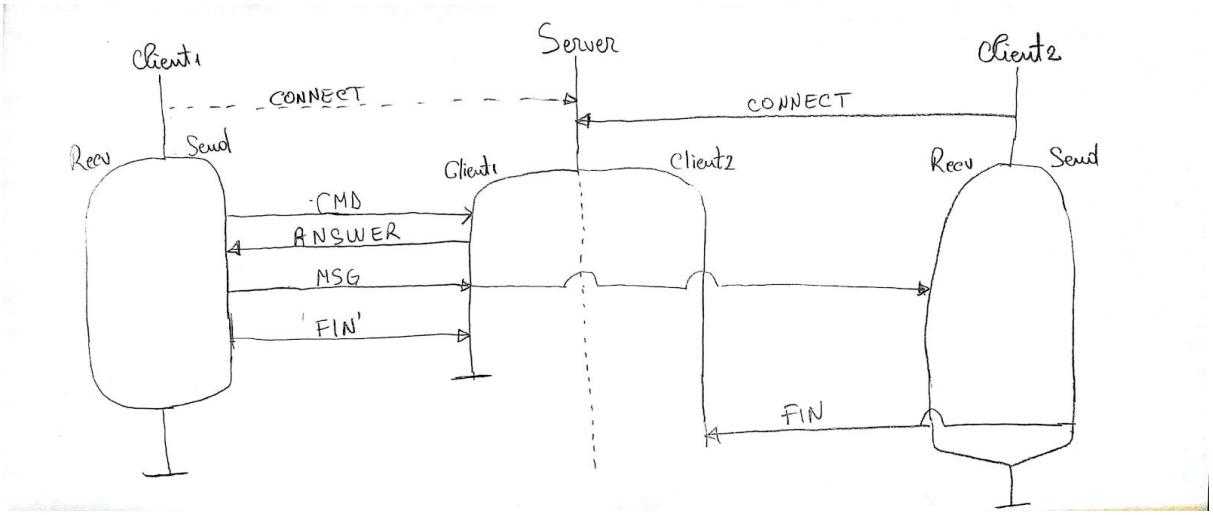
Marian Aldescu
Florine Pratlong

Projet FAR - Livrable 4

Application de messagerie instantanée

I) Protocol



Le Serveur est lancé en exécution et reçoit comme paramètres le porte, l'IP, le nombre de salons et le nombre de clients par un salon.

Son premier tâche est d'attendre les connections des clients et pour chaque nouvelle connection, il crée un thread à recevoir les messages d'un client.

Le Client demande la connexion au serveur, puis le serveur accepte ou refuse et lui transmet la réponse en lui disant qu'il attend qu'au moins un autre client se connecte.

Chaque client crée 2 threads, 1 thread à réceptionner des messages et l'autre à envoyer des messages.

Le Client peut envoyer les messages:

\cmd - lister tous les commandes
\list - lister les détails des salon
\join - ajouter un client dans un salon
\exit - effacer un client d'un salon
\add_c - créer un nouveau salon
\edit_c - éditer les détails d'un salon
\rm_c - effacer un salon

A un moment donné, un client peut être inscrit dans un seul salon. Pour envoyer des messages dans les autres salon, il doit premièrement sortir et puis s'inscrire dans un autre salon.

Pour chaque commande envoyée au serveur, le client reçoit un réponse.

Tous les autres message qui ne sont pas des commandes seront transmettre aux clients dans le même salon avec le client.

Quand le serveur a reçu le message 'fin', il efface les données du client et finit son thread. Le client finit sa execution.

II)

L'architecture d'application:

Le program serveur est premièrement constitué par un thread qui attend les connections des clients. Pour chaque client connecté, il crée un thread pour gérer les messages.

Le program client est constitué par 1 thread dans son premier état, et après il se connecte au serveur, 2 threads sont créés, 1 pour la réception et 1 pour l'envoie des messag

Les ressources partagées par les threads sont déclarées comme variables globales.

On a eu besoin de synchronisation au serveur pour modifier les information des salons, don pour chaque salons il y a un mutex. Pour ajouter/effacer des salons/clients, il y a 2 mutex, un pour la liste de clients et 1 pour le tableau de salons.

III) Nous n'avons pas rencontrés des difficultés importantes.

IV)) Marian c'est occupé de faire le code du serveur, Florine c'est occupée de faire le code du client.

v) Compiler et exécuter le code

Nous avons écrit un fichier Makefile pour automatiser la compilation et l'execution.

Aussi, si on veut changer

l'adresse IP,
le port,
le nombre de salons
le nombre de clients dans un salon

on modifie dans ce fichier les variables:

IP_SERVER,
PORT,
NB_CHANNEL,
NB_CLI.

Pour compiler:

make

ou

make client
make server

Pour tester:
On lance premierement un serveur avec:

```
make run_server
```

puis, on lance clients avec:

```
make run_client
```

Pour effacer les executables:

```
make clean
```

Pour implémenter la liste, nous avons consulté:
https://www.tutorialspoint.com/data_structures_algorithms/linked_list_program_in_c.htm