

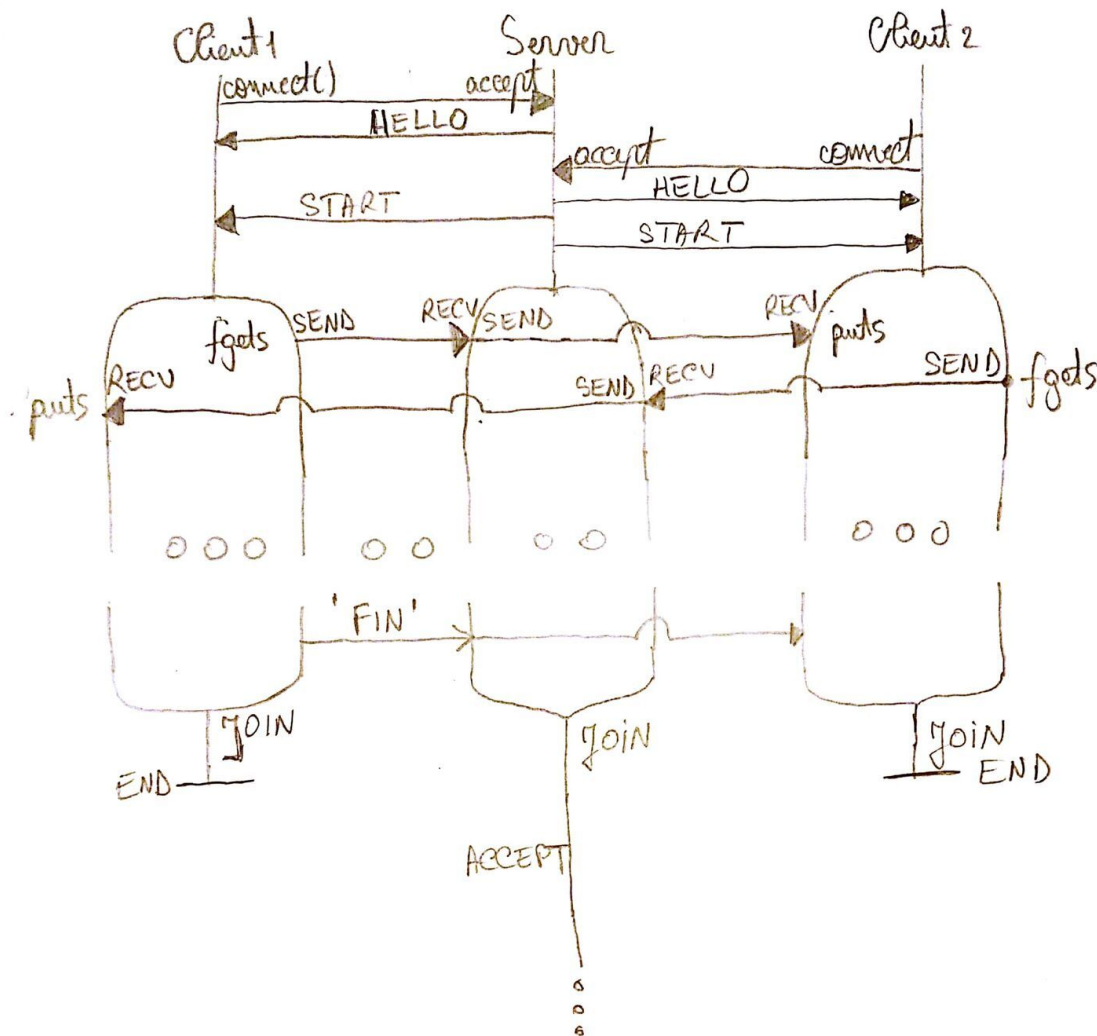
Marian Aldescu
Florine Pratlong

Projet FAR - Livrable 2

Application de messagerie instantanée

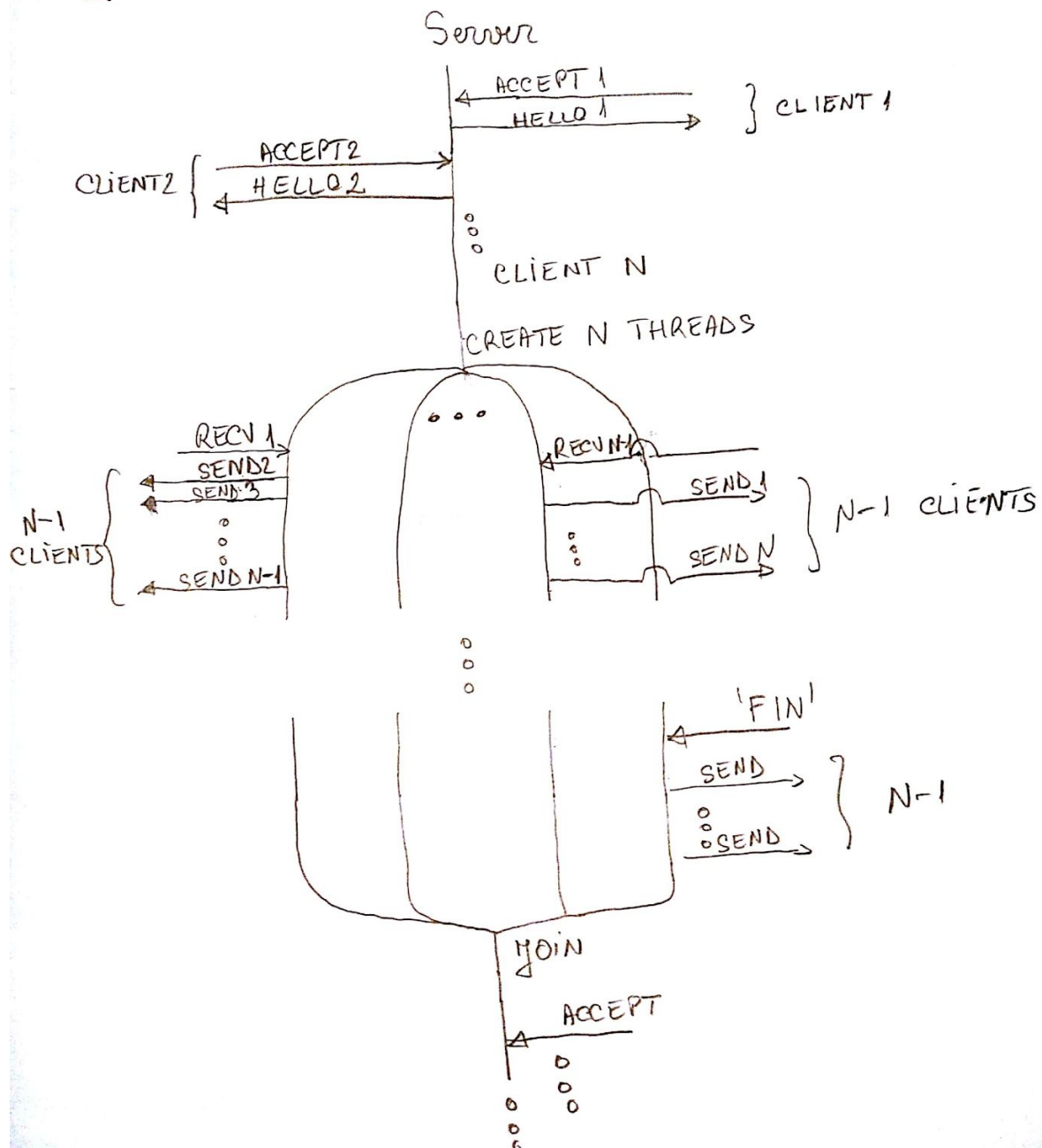
I) Protocol
V1

V1:



V1) le client1 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. Le client2 demande l'autorisation au serveur, le serveur accepte ou refuse et transmet la réponse à ce client. Ensuite il prévient les clients que la communication peut démarrer et les deux clients peuvent s'envoyer des messages. Quand le serveur a reçu le message 'fin', il l'envoie et puis fini les 2 threads. Le serveur attend des autres connexions.

V2



Le serveur attend N connection des clients, et pour chaque client connecte, il envoie un message 'Hello' . Après, il cree N threads, un pour chaque client. Les threads fait la meme chose comme la version 1.

II)

V1.

Le serveur et le client sont constitués par deux processus(chaque).

Dans les 2 programmes(client et serveur) un thread fait **recv** et l'autre thread fait **send**, et comme la socket est **full duplex**, on n'a pas besoin de synchronisation.

V2. Pour la version 2 du serveur on a eu besoin de synchronisation les threads parce que plusieurs threads peuvent écrire sur la même socket une fois. Donc on a fait **send()** dans une zone critique en utilisant une mutex.

Avant d'accéder la socket, c'est exécuté:

pthread_mutex_lock()

et ensuite

pthread_mutex_unlock()

III) Nous avons rencontrés des problèmes pour le V2, quand on a organisé l'information du chaque client dans le serveur_v2. On a fait un plan de modulariser le code pour passer ce problème.

IV) Marian c'est occupé de faire le code des serveurs, 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'exécution.

Aussi, si on veut changer l'**adresse IP**, le **port** ou le **nombre de clients** du serveur, on modifie dans ce fichier les deux variables: **PORT**, **IP_SERVER**, **NB_CLIENTS**.

Pour compiler:

`make`

ou

`make client`

`make server`

`make server_v2`

Pour tester:

V1: On lance premièrement un serveur avec:

`make run_server`

puis, on lance 2 clients avec:

`make run_client`

V2: On lance premierement un serveur avec:

`make run_server_v2`

puis, on lance N clients avec:

`make run_client`

ou N est le nombre de clients qui se connectent au serveur. On peut changer N dans le fichier Makefile. Pour commencer, N = 3.

Pour effacer les executables:

`make clean`