RAPPORT

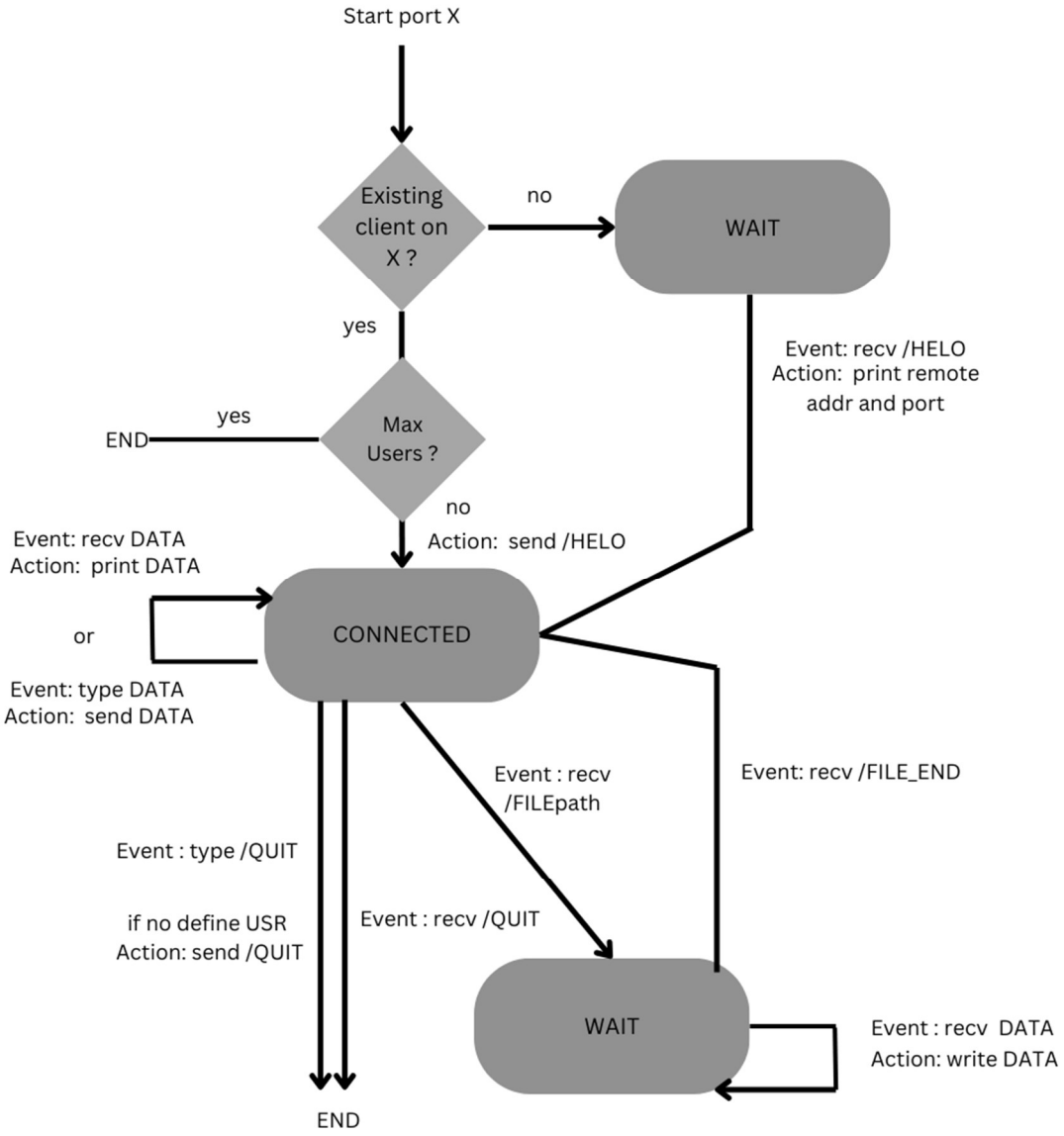
1. Introduction

J'ai fait les fonctionnalités suivantes :

- Protocole dans sa version basique
- Transfert de fichiers
- N utilisateurs

Le transfert de fichier n'utilise rien de nouveau.
La gestion des utilisateurs nécessite de partager de la mémoire, j'ai utilisé shm_open et mmap.
La compilation conditionnelle fonctionne pour n'importe quelle combinaison entre 4/5.
J'ai essayé de commenter le code au maximum.

2. Machine à états



Pour N utilisateurs : On vérifie qu'on ne dépasse pas MAX_USERS, au lieu de send DATA à 1 seul utilisateur, on envoie à tous les autres utilisateurs. On ne send pas QUIT quand on type QUIT.

4. Transfert de fichier

Macro : FILEIO

Détection : /FILE, format des messages identique au protocole de base.

Fonction **sendFile** : extrait le chemin, ouvre et lit le fichier, envoie en continu des messages dans le socket tant qu'il y en a et envoie /FILE_END quand le fichier est entièrement lu.

Fonction **receiveFile** : Ouvre un fichier, écrit les messages reçus depuis le socket dans le fichier, informe l'utilisateur de la fin.

Dans l'algorithme principal, on détecte /FILE puis on appelle les fonctions désignées.

5. Gestion avancée des Utilisateurs

Macro : USR

MAX_USERS à 10.

J'utilise un segment de mémoire partagé pour partager à tous les processus les clients connectés, leur id et leur structure. Je stocke également le nombre d'utilisateurs connectés. Cela permet facilement de partager des informations entre toutes les instances.

A chaque nouvelle connexion, j'ajoute le client à la structure et je modifie le segment en conséquence. J'utilise un flexible array member de taille maximum MAX_USERS.

Grâce à ça, si un client écrit un message dans le terminal je peux faire un broadcast via broadcastMessage : prend en argument le segment partagé, l'index du client qui appelle et le message à envoyer. Puis je boucle sur la variable nb_connected pour envoyer le message à tous les clients excepté lui-même.

6. Conclusion

La gestion des utilisateurs m'a posé pas mal de difficultés au niveau de la compilation conditionnelle. J'ai commencé la partie mode binaire mais je me suis rendu compte que je ne comprenais pas vraiment ce qui était demandé.

L'UE ASE a été utile pour pouvoir utiliser de la mémoire partagée dans ce projet.