

## Un protocole de discussion libre et non fiable...

On veut implémenter un protocole permettant à plusieurs utilisateurs de discuter rapidement et librement (sans authentification) entre eux. Chaque utilisateur utilise un programme client qui envoie un paquet UDP à un serveur contenant son nom (le nom du compte) et le message qu'il envoie. Le serveur renvoie chaque paquet reçu à tous les clients (sauf l'expéditeur) qui l'affichent à leur utilisateur.

- 1) Écrire le client implémentant ce protocole. La fonction `uid_t geteuid(void)` renvoie l'uid du compte courant, on obtient le nom du compte en appelant la fonction `uid_t struct passwd *getpwuid(uid_t uid)`
- 2) Écrire un serveur multi-threads implémentant ce protocole.
- 3) Modifier le serveur de façon à enregistrer chaque message reçu dans un fichier. Chaque message est enregistré sur une ligne qui contient : l'adresse du client, le port du client, la date de réception du message, le nom de l'utilisateur et le message. Plusieurs messages peuvent arriver en même temps, il faut verrouiller l'accès au fichier pendant une écriture.
- 4) Modifier le serveur afin qu'il envoie l'historique de la discussion (les trente derniers messages, envoyés chacun dans son paquet UDP) à chaque nouveau client. Il faut prendre le verrou avant de lire le fichier.