

Systèmes et réseaux

2^{ème} année licence informatique

Projet

Application de messagerie instantanée

But : Le but de ce projet est de créer une application simple de messagerie instantanée en utilisant les sockets selon le modèle client/serveur. Cette application doit permettre à deux utilisateurs d'échanger des messages via un serveur qui communique par sockets TCP/IP.

Partie I- Implémenter le serveur selon le modèle TCP :

Dans un mode connecté, le serveur établit une connexion avec le client. Après la création de la socket et l'établissement de la connexion, il peut recevoir les messages des clients.

- 1) Dans un fichier "serveur.py", créez une socket pour le serveur.
- 2) Associez une adresse et un numéro de port à la socket du serveur.
- 3) Fixez le nombre de connexions acceptées à deux, pour que seulement deux utilisateurs peuvent communiquer via ce serveur.
- 4) Pour la fonction de dialogue, on suppose que le serveur peut envoyer et recevoir des messages d'une manière simultanée (utiliser les threads). Aussi, le serveur et le client doivent pouvoir échanger les données à tout moment et infiniment tant que la connexion est encore établie.
 - A. Écrire les instructions qui permettent au serveur de recevoir un message d'un client.
 - B. Afficher le message reçu sur le terminal avec le nom du client.
 - C. Maintenant c'est le tour du serveur pour transférer le message au deuxième client. Écrire les instructions qui permettent au serveur d'envoyer un message à une station client.
- 5) Écrire une instruction qui permet la fermeture de la socket une fois l'échange des messages est terminé.
- 6) Modifiez le programme de manière à ce que la connexion se termine lorsqu'un des deux clients envoie le texte "quitter". Avant de terminer le programme, notifiez l'autre client que la discussion est terminée.

Partie II- Implémenter le client selon le modèle TCP :

L'adresse IP et le numéro de port du serveur sont obligatoires pour établir une connexion avec le serveur. Ecrivez un fichier "client.py" en suivant les étapes suivantes :

- 1) Créez la socket du client et envoyez une demande de connexion au serveur.
- 2) En utilisant le multithreading, créez deux fonctions qui s'exécutent en parallèle :
 - A. Une fonction qui demande au client de taper un message, et puis l'envoie au serveur.
 - B. Une deuxième fonction qui reçoit les messages du serveur.
- 3) Écrire une instruction qui permet la fermeture de la socket une fois l'échange des messages est terminé.
- 4) Modifiez le programme de manière à ce que la socket soit fermée lorsque le client envoie le message "quitter" ou lorsqu'il reçoit une notification de terminaison de connexion.

Partie III- Interfaces graphiques :

1. Réalisez une interface utilisateur avec trois composants principaux:
 - Une zone d’affichage: qui affiche les messages reçus et envoyés (afficher la date et heure de l’envoi pour chaque message).
 - Une zone de saisie: qui permet au client de saisir son message.
 - Un bouton: qui permet de quitter la discussion (ce bouton ne ferme pas la fenêtre).
2. Modifiez le programme pour qu’au lieu d’envoyer le texte “*quitter*”, l’utilisateur peut quitter la discussion en cliquant sur ce bouton.
3. Affichez un message “*le client X a quitté la discussion.*” et empêchez la saisie de nouveaux messages.

PS: Vous avez le libre choix de la bibliothèque graphique GUI.