**Documentation pour Application de Chat Client-Serveur (Serveur)**

Ce document fournit une documentation pour la partie serveur d'une application de chat client-serveur écrite en C. L'application permet à plusieurs clients de se connecter simultanément et d'échanger des messages à travers le serveur. Chaque client est identifié par un pseudonyme unique. Le serveur gère les connexions, les déconnexions, et la diffusion des messages à tous les clients connectés.

**Implémentation**

**Dépendances**

* Bibliothèques standard C : stdio.h, stdlib.h, string.h, unistd.h
* Bibliothèques réseau : arpa/inet.h
* Bibliothèques de threads : pthread.h
* Bibliothèques de gestion de fichiers : fcntl.h
* Bibliothèques d'appels système : sys/types.h, sys/stat.h
* Bibliothèques de gestion du temps : time.h

**Constantes**

* MAX\_CLIENTS : Définit le nombre maximal de clients que le serveur peut gérer simultanément.
* BUFFER\_SIZE : Définit la taille du tampon de message pour l'envoi et la réception de messages.

**Structures de Données**

* UserInfo : Structure contenant les informations sur un utilisateur (adresse IP, pseudonyme, socket).

**Variables Globales**

* users : Tableau de structures UserInfo contenant les informations sur tous les utilisateurs connectés.
* userCount : Variable indiquant le nombre actuel d'utilisateurs connectés.
* userMutex : Mutex pour assurer une manipulation sûre de la table des utilisateurs par plusieurs threads.

**Fonctions**

**main**

La fonction principale du serveur gère la création du socket, la configuration de l'adresse du serveur, la liaison du socket à l'adresse et au port, la mise en mode écoute, et l'acceptation des connexions entrantes. Chaque nouvelle connexion est gérée par un thread distinct (clientThread).

**clientThread**

Ce thread est responsable de la gestion d'une connexion client individuelle. Il appelle la fonction handleConnection pour gérer l'établissement initial de la connexion et ensuite gère la réception des messages.

**handleConnection**

Cette fonction est appelée lorsqu'un client établit une connexion. Elle vérifie la disponibilité du pseudonyme du client, l'ajoute à la table des utilisateurs, envoie un message de bienvenue au client, et gère les messages entrants jusqu'à la déconnexion du client.

**handleMessage**

Cette fonction est appelée pour traiter les messages reçus d'un client. Elle formate le message avec l'heure, l'adresse IP, le pseudonyme du client, et le message lui-même. Ensuite, elle envoie le message à tous les clients connectés, l'enregistre dans le fichier chat.txt, et l'affiche dans le terminal du serveur.

**Gestion des Erreurs**

* Le serveur fournit des messages d'erreur en cas de problèmes lors de la création du socket, la liaison, la mise en mode écoute, et la gestion des connexions.

**Nettoyage**

* Le serveur libère les ressources en fermant le socket avant de se terminer.