

TP3 : LES SOCKETS

Exercice 1 : Affichage de la date et de l'heure et la date d'un serveur depuis une machine cliente à un à l'aide de Sockets TCP.

DESCRIPTION:

Le serveur TCP obtient la date et l'heure du système et ouvre le socket du serveur pour lire le détail du client. Le client envoie son requête au serveur. Ensuite, le client reçoit la date et l'heure du serveur à afficher. La connexion client du serveur de socket TCP est ouverte pour la communication. Une fois l'heure affichée, la connexion client-serveur est fermée et ses flux respectifs doivent être fermés.

A. Serveur

1. Créez un socket de serveur et associez-le au port.
2. Écoutez la nouvelle connexion et, quand une connexion arrive, acceptez-la.
3. Envoyez la date et l'heure du serveur au client.
4. Lisez l'adresse IP du client envoyée par le client.
5. Affichez les détails du client.
6. Répétez les étapes 2 à 5 jusqu'à ce que le serveur soit terminé.
7. Fermez tous les flux.
8. Fermez le socket du serveur.

B. Client

1. Créez un socket client et connectez-le au numéro de port du serveur.
2. Récupérez sa propre adresse IP en utilisant la fonction intégrée.
3. Envoyez son adresse au serveur.
4. Affichez la date et l'heure envoyées par le serveur.
5. Fermez les flux d'entrée et de sortie.
6. Fermez le socket client.

Exercice 2 : Communication client-serveur à l'aide de sockets TCP

DESCRIPTION:

Les clients TCP envoient une demande au serveur et le serveur recevra la demande et réponse au client avec accusé de réception. Chaque fois que le client communique avec le serveur, le serveur lui répond.

A. Serveur

1. Créez un socket de serveur et associez-le au port.
2. Écoutez la nouvelle connexion et, quand une connexion arrive, acceptez-la.
3. Lire le message du client et l'afficher
4. Recevez un message de l'utilisateur et envoyez-le au client
5. Répétez les étapes 3 et 4 jusqu'à ce que le client envoie "end"
6. Fermer tous les flux
7. Fermez le serveur et le socket client

B. Client

1. Créez un socket client et connectez-le au numéro de port du serveur.
2. Recevez un message de l'utilisateur et envoyez-le au serveur
3. Lire la réponse du serveur et l'afficher

4. Répétez les étapes 2 et 3 jusqu'à ce que le chat se termine avec le message "end"
5. Fermer tous les flux d'entrée / sortie
6. Fermer le socket client

Exercice 3 : Mise en oeuvre de TCP / IP echo à l'aide de sockets TCP

DESCRIPTION:

Le serveur TCP reçoit le message et ouvre le socket du serveur pour lire les détails du client. Le client envoie son adresse au serveur. Ensuite, le client reçoit le message du serveur à afficher.

A. Serveur

1. Créez un socket de serveur et associez-le au port.
2. Écoutez la nouvelle connexion et, quand une connexion arrive, acceptez-la.
3. Lisez les données du client.
4. Renvoyez les données au client.
5. Répétez les étapes 4 à 5 jusqu'à ce que «bye» ou «null» soit lu.
6. Fermez tous les flux.
7. Fermez le socket du serveur.

B. Client

1. Créez un socket client et connectez-le au numéro de port du serveur.
2. Obtenez les commentaires de l'utilisateur.
3. Si égal à bye ou null, passez à l'étape 7.
4. Envoyez les données utilisateur au serveur.
5. Affichez les données renvoyées par le serveur.
6. Répétez les étapes 2 à 4.
7. Fermez les flux d'entrée et de sortie.
8. Fermez le socket client.

Exercice 4 : Programme chat server/client utilisant socket UDP

DESCRIPTION:

UDP est un protocole sans connexion et le socket est créé pour que le client et le serveur transfèrent les données. La connexion de socket est réalisée en utilisant le numéro de port.

Serveur

1. Créez deux ports, port du serveur et port du client.
2. Créez un socket de datagramme et associez-le au port client.
3. Créez un paquet de datagramme pour recevoir le message du client.
4. Attendez les données du client et acceptez-les.
5. Lire le message du client.
6. Obtenir les données de l'utilisateur.
7. Créez un paquet de datagramme et envoyez un message via le port du serveur.
8. Répétez les étapes 3 à 7 jusqu'à ce que le client ait quelque chose à envoyer.
9. Fermez le socket du serveur.

Client

1. Créez deux ports, port du serveur et port du client.
2. Créez un socket datagramme et associez-le au port du serveur.

3. Obtenir les données de l'utilisateur.
4. Créez un paquet de datagramme et envoyez des données avec l'adresse IP du serveur et le port client.
5. Créez un paquet datagramme pour recevoir le message du serveur.
6. Lisez la réponse du serveur et affichez-la.
7. Répétez les étapes 3 à 6 jusqu'à ce qu'il y ait du texte à envoyer.
8. Fermez le socket client.

Exercice 5 : Implémentation du SERVER / CLIENT DNS à l'aide de sockets UDP

DESCRIPTION :

DNS signifie système de nom de domaine. Le nom unique de un hôte est identifié par son IP adresse via la communication client serveur.

Serveur

1. Créez un tableau d'hôtes et son adresse ip dans un autre tableau
2. Créer un socket datagramme et le lier à un port
3. Créer un paquet de datagramme pour recevoir la demande du client
4. Lisez le nom de domaine du client à résoudre.
5. Recherchez le tableau d'hôte pour le nom de domaine.
6. Si trouvé, récupérez l'adresse correspondante
7. Créez un paquet de datagramme et envoyez une adresse IP au client
8. Répétez les étapes 3 à 7 pour résoudre les demandes supplémentaires des clients.
9. Fermez le socket du serveur

Client

1. Créer un socket de datagramme
2. Obtenir le nom de domaine de l'utilisateur
3. Créez un paquet de datagramme et envoyez le nom de domaine au serveur
4. Créez un paquet de datagramme pour recevoir le message du serveur.
5. Lire la réponse du serveur
6. Si l'adresse IP ne s'affiche pas, affichez "Le domaine n'existe pas"
7. Fermer le socket client

Exercice 6 : Capture et filtrage de paquets en utilisant socket raw

DESCRIPTION:

Le socket Raw est créé pour définir la transmission du paquet. Le paquet est capturé pour vérification d'erreur. Les erreurs contenant des paquets sont filtrées pendant la transmission. Un socket Raw utilisant le protocole TCP / IP est créée. La longueur du paquet est définie avec l'en-tête TCP associé. Les paquets avec erreur sont vérifiés en utilisant le mécanisme CRC et filtrés.

Programme:

1. Lancer le programme et inclure les fichiers d'en-tête nécessaires.
2. Définir la longueur du paquet.
3. Déclarer la structure d'en-tête IP à l'aide de l'en-tête TCP.
4. Utiliser un processus de somme de contrôle simple pour vérifier le processus.
5. Utilisation du protocole de communication TCP \ IP pour exécuter le programme.
6. Et en utilisant la communication TCP \ IP pour entrer l'adresse IP source, le numéro de port, l'adresse IP cible et le numéro de port.