

Rapport de mini-projet relatif au module Programmation C sous Unix

Barouni Abir

I. Introduction :

Dans cette présentation, nous allons examiner le travail accompli pour développer un système de communication client-serveur utilisant à la fois le protocole *UDP (User Datagram Protocol)* dans la première partie et le protocole *TCP (Transmission Control Protocol)* pour la deuxième partie.

II. Section 1 : Implémentation de la communication client-serveur avec UDP :

Dans cette section, nous détaillerons l'implémentation de la communication client-serveur en utilisant le protocole UDP.

Le protocole UDP est un protocole de transport léger et sans connexion, adapté aux applications nécessitant une communication rapide et peu complexe. Ce sont les étapes nécessaires pour établir une communication UDP entre le client et le serveur :

1.1 Établissement de la connexion UDP :

Les clients peuvent envoyer des datagrammes UDP au serveur sans établir une connexion préalable.

1.2 Envoi et réception de datagrammes UDP :

Les clients envoient des datagrammes UDP au serveur, en spécifiant l'adresse IP et le numéro de port du serveur. Le serveur, de son côté, sera capable de recevoir les datagrammes UDP et de les traiter.

1.3 Limite de la connexion UDP :

En effet, le protocole UDP ne garantit pas la remise des datagrammes ni ne gère les problèmes de perte ou de réorganisation des paquets. Contrairement au protocole TCP, qui offre une transmission fiable, UDP traite les datagrammes de manière indépendante et n'effectue pas de retransmission automatique en cas de perte ou de corruption de données. Cela signifie que certains datagrammes UDP peuvent ne pas être reçus par le destinataire ou être reçus dans un ordre différent de celui dans lequel ils ont été envoyés.

III. Section 2 : Implémentation de la communication client-serveur avec TCP :

Dans cette section, nous détaillerons l'implémentation de la communication client-serveur en utilisant le protocole TCP (Transmission Control Protocol). Contrairement à UDP, TCP est un protocole orienté connexion et fiable, adapté aux applications nécessitant une transmission de données sans perte et dans l'ordre.

2.1 Établissement de la connexion TCP :

Le processus d'établissement de la connexion TCP comprend une poignée de main à trois volets entre le client et le serveur, et peut également inclure l'authentification du client à l'aide d'un nom d'utilisateur et d'un mot de passe.

Lorsque le client tente de se connecter au serveur, il fournit un nom d'utilisateur et un mot de passe associés à son compte. Le serveur reçoit ces informations et procède à la vérification des informations d'identification fournies par le client.

- Si les informations d'identification fournies par le client sont valides, le serveur accepte la connexion TCP et continue le processus d'établissement de la connexion et si les informations d'identification sont incorrectes il peut refuser la connexion et envoyer un message d'erreur au client.

2.2 Envoi et réception de données avec TCP :

Nous décrirons comment les clients envoient des données au serveur à l'aide de connexions TCP établies.

➤ Client

1. Le client se connecte au serveur TCP.
2. Le client affiche un menu à l'utilisateur, lui présentant différentes options.
3. L'utilisateur sélectionne une option dans le menu.
4. Le client envoie la sélection au serveur via la connexion TCP établie.
5. Le client attend la réponse du serveur.

➤ Serveur

1. Le serveur écoute les connexions entrantes sur un certain port TCP.
2. Lorsqu'une connexion est établie avec un client, le serveur accepte la connexion.
3. Le serveur attend de recevoir les données du client.
4. Le serveur traite les données reçues pour déterminer quelle action effectuer.
5. Le serveur génère la réponse correspondante en fonction de l'action choisie.
6. Le serveur envoie la réponse au client via la connexion TCP établie.

2.3 Problèmes rencontré :

Nous avons rencontré des problèmes lors de l'utilisation de la bibliothèque *GTK+* pour créer une interface graphique.

Malgré les fonctionnalités puissantes qu'elle offre, l'inclusion de **#include <gtk/gtk.h>** a posé des difficultés dans notre processus de développement.

IV. Conclusion

En conclusion, ce rapport a présenté l'implémentation de la communication client-serveur en utilisant les protocoles UDP et TCP. Le protocole UDP offre une communication légère et rapide, mais moins fiable, tandis que le protocole TCP assure une transmission fiable et ordonnée des données.