

Rapport de Projet SE

Système de Messagerie en Réseau

M'BASSIDJE Timothée et NICOLAS Ethan

13 mai 2024

1 Introduction

Ce rapport présente le développement d'un système de messagerie réseau basé sur le protocole TCP/IP. Le projet est implémenté en langage C, utilisant les bibliothèques standards Unix pour la gestion des opérations système et réseau. Ce système comprend des composants serveur et client, offrant des fonctionnalités complètes pour l'envoi et la réception de messages entre les utilisateurs, ainsi qu'une interface GTK dédiée.

Les fichiers sources incluent des définitions pour les serveurs et les clients, ainsi que pour une interface utilisateur graphique qui facilite l'interaction avec le système. Le projet est structuré pour permettre une communication efficace et sécurisée, en mettant en œuvre des fonctions de connexion réseau, de gestion des utilisateurs et des sessions, et de traitement des données transmises.

1.1 Technologies Utilisées

Le projet utilise les technologies suivantes :

- Langage C.
- Bibliothèques C pour les opérations système et réseau TCP.
- Makefile pour l'automatisation de la compilation.
- GTK3 pour l'interface utilisateur

2 Explications du Code

2.1 Serveur TCP

```
// Exemple de fonction principale du serveur
int main(int argc, char *argv[]) {
    int sock_ecoute = creer_configurer_sock_ecoute(PORT_WCP);
    if (sock_ecoute < 0) {
        perror("Erreur_de_creation_de_la_socket_d'ecoute");
        exit(EXIT_FAILURE);
    }
    while (1) {
        int fd_client = accepter_connexion(sock_ecoute);
        if (fd_client < 0) continue;
        pthread_t thread_id;
        if (pthread_create(&thread_id, NULL, thread_worker, (void *)&fd_client) != 0) {
            perror("Erreur_de_creation_du_thread");
        }
    }
    return 0;
}
```

2.2 Client TCP

```
// Exemple de fonction pour envoyer des requetes au serveur
void envoyer_query(int fd, query_t *q) {
    if (write(fd, q, sizeof(*q)) < 0) {
```

```

        perror("Erreur_d'envoi_de_la_requete");
        exit(EXIT_FAILURE);
    }
}

```

2.3 Interface Utilisateur GTK

```

// Exemple de fonction pour initialiser l'interface utilisateur
void init_ui() {
    GtkWidget *window = gtk_window_new(GTK_WINDOW_TOPLEVEL);
    gtk_window_set_title(GTK_WINDOW(window), "Messagerie_TCP");
    gtk_window_set_default_size(GTK_WINDOW(window), 350, 200);
    gtk_widget_show_all(window);
}

```

3 Conclusion

Ce projet a démontré l'efficacité de l'utilisation de C et des sockets TCP/IP pour développer un système de messagerie réseau robuste. Malgré les défis liés à la gestion des communications réseau et des interfaces utilisateur en C, le système accompli permet une communication fluide et sécurisée entre les utilisateurs. Les perspectives futures pourraient inclure l'amélioration de l'interface utilisateur, l'ajout de chiffrement pour les messages, et une meilleure gestion des erreurs et des performances réseau.