

Rapport de Projet

Système de Discussion Utilisant des Tubes Nommés

Institut Galilée

Année 2022-2023

ABDOULAYE BARRY

Encadrant : M. Kais

Contents

1	Introduction	2
2	Approche de Solution	2
2.1	Architecture du Système	2
2.2	Gestion des Utilisateurs	2
2.3	Communication Interprocessus	2
2.4	Synchronisation et Gestion des Messages	2
2.5	Interface Utilisateur	2
3	Implémentation	3
3.1	Création des Tubes Nommés	3
3.2	Synchronisation et Gestion des Messages	3
4	Problèmes Rencontrés	3
4.1	Gestion des Tubes Nommés	3
4.2	Synchronisation des Processus	3
5	Réflexions et Améliorations Potentielles	3
5.1	Limitations des Tubes Nommés	3
5.2	Améliorations Proposées	4
6	Conclusion	4

1 Introduction

Ce rapport présente le projet de développement d'un système de discussion utilisant des tubes nommés. L'objectif de ce projet est de créer un programme permettant à plusieurs utilisateurs de communiquer en temps réel via des canaux de discussion.

Le système de discussion offre une plateforme virtuelle où les utilisateurs peuvent participer à des discussions simultanées, échanger des messages et interagir les uns avec les autres. Ce projet a été réalisé en utilisant le langage C, en tirant parti des fonctionnalités du système d'exploitation pour la gestion des tubes nommés.

Dans ce rapport, nous présenterons les différentes étapes de conception et d'implémentation du système, les problèmes rencontrés, les solutions apportées ainsi que les améliorations potentielles et les limites du système.

2 Approche de Solution

Dans cette section, nous décrivons notre approche de solution pour la mise en œuvre du système de discussion utilisant des tubes nommés.

2.1 Architecture du Système

Nous avons opté pour une architecture client-serveur. Le serveur agit comme un point centralisé de communication, tandis que les clients se connectent au serveur pour participer aux discussions. Cette architecture permet une gestion centralisée des connexions, la coordination des échanges de messages et la gestion des utilisateurs.

2.2 Gestion des Utilisateurs

Pour gérer les utilisateurs, nous avons créé une structure de données appelée **User** qui stocke les informations essentielles de chaque utilisateur (nom, statut de connexion, tubes de lecture et d'écriture).

2.3 Communication Interprocessus

La communication est réalisée à l'aide des tubes nommés. Chaque client possède un tube d'écriture et un tube de lecture pour échanger des messages avec le serveur.

```
1 mkfifo(tubeLecture, 0666);  
2 mkfifo(tubeEcriture, 0666);
```

Listing 1: Extrait de la création des tubes nommés

2.4 Synchronisation et Gestion des Messages

Pour assurer une communication fluide et synchronisée, des verrous (mutex) et des variables de condition sont utilisés pour contrôler l'accès aux tubes nommés et garantir l'exclusivité de l'écriture.

2.5 Interface Utilisateur

Nous avons développé une interface en ligne de commande pour permettre aux utilisateurs de saisir et de recevoir des messages. Les commandes spéciales telles que `/quit` permettent de se déconnecter.

3 Implémentation

3.1 Création des Tubes Nommés

L'implémentation commence par la création des tubes nommés. La fonction `mkfifo` est utilisée pour créer ces tubes, qui sont les canaux de communication entre le serveur et les clients.

```
1 typedef struct {  
2     char nomUtilisateur[MAX_USERNAME_LENGTH];  
3     int estConnecte;  
4     int tubeLecture;  
5     int tubeEcriture;  
6 } User;  
7 User utilisateurs[MAX_USERS];
```

Listing 2: Structure `User` pour la gestion des utilisateurs

3.2 Synchronisation et Gestion des Messages

Pour éviter les conflits, des verrous sont utilisés lors de l'écriture dans les tubes. Voici un extrait montrant l'utilisation d'un verrou pour écrire dans un tube :

```
1 pthread_mutex_lock(&verrouTubeEcriture);  
2 write(tubeEcritureClient, message, strlen(message) + 1);  
3 pthread_mutex_unlock(&verrouTubeEcriture);
```

Listing 3: Utilisation des verrous pour synchroniser l'écriture

4 Problèmes Rencontrés

4.1 Gestion des Tubes Nommés

L'un des principaux problèmes était de s'assurer que les tubes étaient créés avant le démarrage du serveur et des clients. Nous avons mis en place un mécanisme de synchronisation pour garantir la bonne création et suppression des tubes nommés.

4.2 Synchronisation des Processus

Lors de l'envoi simultané de messages par plusieurs clients, il était essentiel de garantir un accès exclusif aux tubes nommés. Des verrous (mutex) ont été utilisés pour synchroniser ces opérations.

5 Réflexions et Améliorations Potentielles

5.1 Limitations des Tubes Nommés

Les tubes nommés présentent des limitations, notamment en termes de portabilité sur différentes plateformes et la complexité de nettoyage après utilisation. Une amélioration pourrait être l'utilisation de sockets pour faciliter la portabilité.

5.2 Améliorations Proposées

Nous pourrions développer une interface graphique pour améliorer l'interaction utilisateur, ajouter des fonctionnalités de sécurité telles que l'authentification, et implémenter des fonctions avancées telles que la création de salles de discussion privées.

6 Conclusion

Ce projet nous a permis de développer un système de discussion utilisant des tubes nommés, et d'acquérir une expérience précieuse en programmation système et communication interprocessus. Nous avons exploré les avantages et les limitations des tubes nommés, et identifié des améliorations potentielles pour rendre notre système plus robuste, sécurisé et convivial.