

Table des matières

1	Application	3
1.1	Architecture	3
1.1.1	Console/Serveur	3
1.1.2	Chatmmap	3
1.2	Conception	4
1.2.1	Console/Serveur	4
1.2.2	Chatmmap	4
2	Tests	5
2.1	Méthodologie	6
2.1.1	Console/Serveur	6
2.1.2	Chatmmap	6
2.2	Exemples	6
2.2.1	Console/Serveur	6
2.2.2	Chatmmap	7
2.3	Difficultés	7
2.4	Solutions	7

Introduction

Dans le cadre du cours *Systèmes d'exploitation centralisés*, réaliser un minichat, basé sur une application de chat classique.

Ce rapport présente mon minichat, avec mes choix de conceptions ainsi que des tests réalisés.

Chapitre 1

Application

1.1 Architecture

1.1.1 Console/Serveur

Ce minichat est basé sur le type serveur/console, c'est-à-dire qu'il y a un processus serveur qui fait le lien entre tous les processus console à travers des tubes nommés.

Donc tout d'abord le serveur est lancé, il crée le tube écoute via lequel toute les connexions se fera. Ensuite les consoles se lancent et se connectent au serveur, elles créent des tubes nommés serveur à console (S2C) et console à serveur (C2S) qui vont servir à envoyer et recevoir des messages.

1.1.2 Chatmmap

Ce minichat n'est basé sur aucune architecture classique : chaque processus sert de serveur et de console à la fois.

L'utilisateur se connecte via un couplage de mémoire à travers un mot clé (nom du serveur). Pour que d'autres utilisateurs rejoignent la même conversation, il leur suffit d'utiliser le même nom de serveur.

Le problème avec cette architecture c'est que le chat n'est pas actualisé à chaque message envoyé.

1.2 Conception

1.2.1 Console/Serveur

J'ai tout d'abord commencé par écrire le code du serveur, sans lequel rien n'est possible. Ensuite j'ai écrit le code des consoles qui se connectent au serveur.

Les détails sont dans les commentaires du code.

1.2.2 Chatmmap

Les détails sont dans les commentaires du code.

Chapitre 2

Tests

2.1 Méthodologie

2.1.1 Console/Serveur

Pour tester l'application console/serveur, j'ai connecté plusieurs utilisateurs à un même serveur en même temps. Ensuite j'essayais d'envoyer des messages, de déconnecter la machine serveur et les machines console via signaux SIGKILL et SIGINT. Des résultats de tests sont disponibles en figure 2.1.

2.1.2 Chatmmap

Idem.

Des résultats de tests sont disponibles en figure 2.2.

2.2.1 Console/Serveur

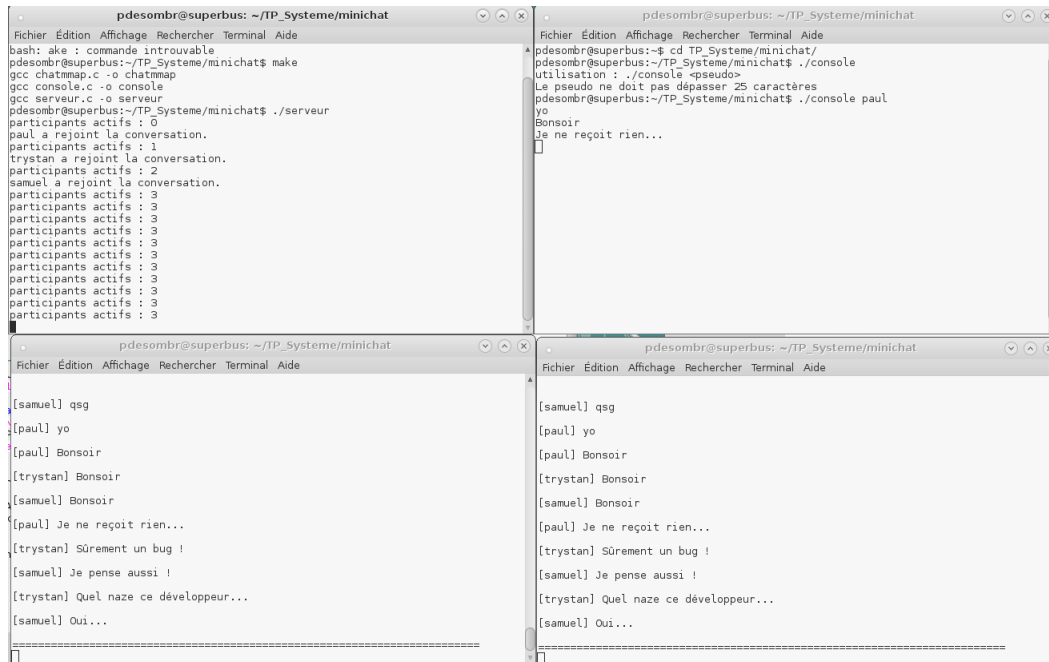


FIGURE 2.1 – Exemple de fonctionnement du minichat console/serveur à trois utilisateurs

2.2.2 Chatmmap

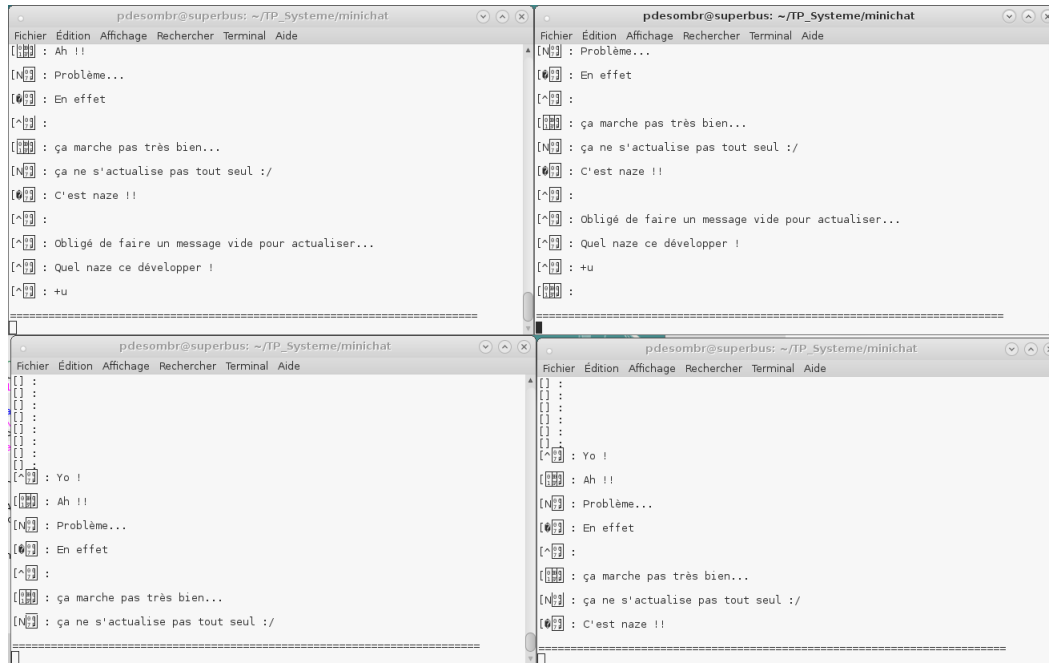


FIGURE 2.2 – Exemple de fonctionnement du minichat chatmmap à quatre utilisateurs

2.3 Difficultés

- Appel de `ISSET()` bloquant (console/serveur) ;
- Dire "au revoir" fait planter le serveur (console/serveur) ;
- Actualisation non automatique du chatmmap ;
- Problème d'affichage des pseudo sur le chatmmap.

2.4 Solutions

- Changer les indices de tableaux ;
- Aucune solution pour la suite.

Conclusion

Pour conclure sur ce projet, je l'ai trouvé intéressant et moins difficile que le minishell.