

Rapport du Projet Terminal



Table des matières

- I. Introduction :2
- II. Compilation et exécution de l'application :3
- III. Fonctionnement de l'application :4
- IV. Choix techniques du code concernant la partie Systèmes Distribués :5
- V. Choix techniques du code concernant la partie Systèmes d'Exploitation :5
- VI. Conclusion :6

I. Introduction :

L'objectif de ce projet est l'implémentation d'un outil de chat dans lequel plusieurs personnes peuvent communiquer par messages.

Dans cette application, un utilisateur disposant d'un compte (mot de passe et d'un pseudo), peut se connecter au système de chat. Un utilisateur connecté peut alors envoyer un message qui sera transmis à l'ensemble des utilisateurs connectés et affiché dans leurs interfaces utilisateur.

L'application est composée de plusieurs parties, écrites et implémentées en java et en c.

II. Compilation et exécution de l'application :

Exécution de l'application :

Dans un premier temps, il est nécessaire d'ouvrir un terminal à la racine du projet, puis, on compile tous les programmes avec les commandes :

```
chmod +x build.sh
```

```
bash build.sh
```

Ensuite, il faut lancer les programmes suivants un ordre précis :

D'abord, gestion-compte.jar inclus dans le dossier gestion-compte avec :

```
java -jar gestion-compte.jar <ip> <port>
```

Puis dans le dossier centre, il faut exécuter le fichier main.o avec la commande :

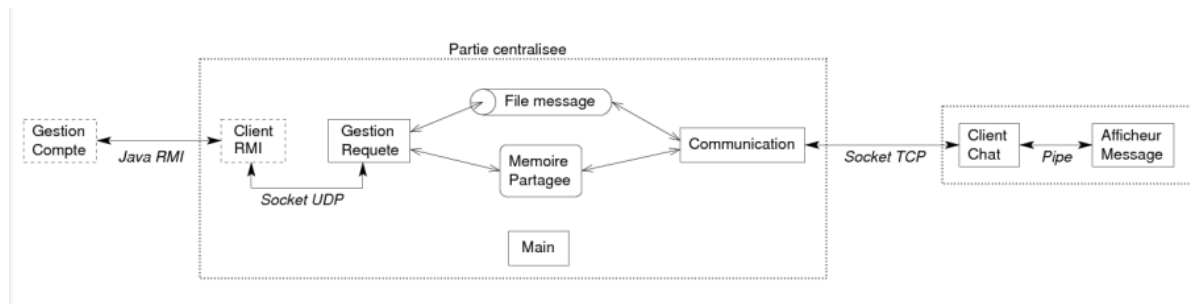
```
./main.o <gestion-compte-ip> <gestion-compte-port> <client-rmi> <communication-port>
```

Puis, dans le dossier client, il faut exécuter le fichier main.o avec la commande suivante :

```
./main.o <communication-ip> <communication-port>
```

III. Fonctionnement de l'application :

Nous avons divisé l'application en trois sous-parties reprenant les parties décrites par l'architecture générale de l'application :



Tout d'abord, la partie gestion Compte, codée en java est présente dans le dossier gestion-compte. Ensuite, la Partie centralisée est codée en c dans le dossier centre (avec la partie Client RMI en java), et enfin, la partie Client Chat, le pipe reliant celle-ci et Afficheur Message, sont présents dans le dossier client. Le socket TCP est codé dans le fichier « main.c » dans le dossier client, et aussi dans « main.c » dans le dossier centre.

Partie Gestion-Compte :

Cette partie permet la gestion des comptes des utilisateurs. Dans le dossier src, le fichier « ICompte.java » comporte l'interface ICompte composée des fonctions register, login et delete. Le fichier « App.java » implémente les différentes fonctions load, save et delete en plus des fonctions de ICompte. La liste des comptes est enregistrée dans le fichier « users.txt ».

Partie Centre :

Dans cette partie, la liaison entre la partie Gestion-Compte et la partie Client est assurée. Le dossier client-rmi contient un fichier comportant aussi l'interface de ICompte nommé « ICompte.java » ainsi que « App.java » qui contient le processus Client RMI et qui génère un socket UDP. Cette partie est de plus, composée du fichier « main.c », élément principal de la partie centralisée. Dans ce fichier se trouve l'implémentation de la mémoire partagée, la gestion de requête, la file de message avec un semaphore et le processus communication.

Partie Client :

Dans cette partie, le fichier « main.c » correspond à Client Chat et « display.c » à Afficheur Message. « display.c » va afficher les messages et les fonctions transmises depuis « main.c » via un pipe. « main.c » définit les fonctions permettant de se connecter, de s'enregistrer, de voir la liste des personnes connectées, d'effacer le profil d'un utilisateur, d'envoyer/recevoir un message et de l'afficher, ainsi qu'un menu permettant à l'utilisateur de choisir une action. De plus, il définit le socket TCP reliant cette partie à la partie centrale.

IV. Choix techniques du code concernant la partie Systèmes Distribués :

Concernant la partie systèmes distribués, dans ce projet, nous avons opté pour :

- L'utilisation de sockets UDP et TCP dans les parties centre et client.
- L'utilisation de RMI.
- L'utilisation de la mémoire partagée dans la partie centre
- L'utilisation de flux.
- L'utilisation de Sockets et de ports

V. Choix techniques du code concernant la partie Systèmes d'Exploitation :

Concernant la partie systèmes distribués, dans ce projet, nous avons opté pour :

- L'utilisation de sémaphore dans la partie centre afin de gérer la cohérence de la file de message.
- L'utilisation de signaux dans la partie gestion-compte afin de fermer proprement les processus.
- L'utilisation de pipes nommés dans la partie client.
- Utilisation de processus et de threads.

VI. Conclusion :

Ce projet nous aura permis de croiser les techniques de codes vues en cours et en TD/TP en Systèmes Distribués et en Systèmes d'Exploitation, afin de réaliser cette application permettant à plusieurs utilisateurs de dialoguer. Grâce à ce projet, nous avons pu appliquer concrètement ces notions. Ce projet aura nécessité beaucoup de rigueur et de réflexion et d'organisation afin de croiser intelligemment les commandes et fonctions vu dans les deux matières, en plus de croiser des connaissances en programmation en langage c et en java.