

Programmation système

Rapport projet 2

Christian Frantzen

Décembre 2014

Contents

Introduction.....	2
Début du projet	2
La partie serveur	2
La partie client	2
Choix d'implémentation.....	3
Défauts de mon projet	3
Conclusion	3

Introduction

Pour le projet 2 en systèmes d'exploitations on devait implémenter une application serveur-client mettant en place un salon de discussion virtuel. Ce salon de discussion est unique et ouvert à tout le monde, c'est-à-dire qu'il n'y a pas de restrictions pour le nombre d'utilisateurs connecté. On se connecte au serveur, choisit un « nickname » et on est prêt à recevoir tous les messages des autres utilisateurs dans le salon. L'envoi de message se fait en console, en tapant le message au clavier, puis on le transmet aux autres utilisateurs.

Début du projet

Ceci étant un projet serveur-client, il y a deux parties : l'implémentation d'un serveur unique qui gère la distribution de messages et celle d'un client utilisable par beaucoup de personnes.

La partie serveur

Pour créer une connexion avec d'autres processus situés sur des machines différentes, il faut initialiser son propre socket. Le port utilisé pour les connexions pour ce projet est le 5555, le serveur doit donc toujours le surveiller. Après avoir initialisé son propre socket, le serveur passe en mode passif où il attend des événements externes.

Lorsqu'il reçoit un message, il doit vérifier si ce message vient d'un processus connu ou pas. Si oui, il continue avec l'envoi de ce message à tous les autres processus connus, si non, il doit ajouter ce nouveau processus à l'ensemble des processus connus.

La partie client

Pour profiter du service, il faut une application client qui se connecte au serveur. Celle-ci initialise donc d'abord son propre socket et par la suite essaie de se connecter au serveur. Notons que l'utilisateur est demandé son « nickname » de préférence pour utiliser le service lors du lancement d'une application client.

Après avoir établie une connexion avec le serveur, l'utilisateur peut commencer à entrer son message.

Choix d'implémentation

Pour chacune des parties du projet, j'ai un fichier pour mon code. Étant bien diviser en fonctions indépendantes, j'ai veillé à l'encapsulation des différentes sections des programmes.

Tous les événements entre les sockets sont gérés par les ensembles de *file descriptors*. N'utilisant pas d'autres structures de données, ceux-ci suffisent pour mon salon de discussion virtuel.

Défauts de mon projet

Dû à des problèmes d'organisation, je n'ai pas su rendre mon projet dans les bons délais. C'est aussi la raison pour laquelle je n'ai pas fait une implémentation adéquate de certains aspects du projet, notamment le stockage des « nicknames » du côté serveur. Ceux-ci sont stockés dans les applications client des utilisateurs. J'aurais du faire un struct renfermant le numéro de socket et le « nickname » de chaque application clients dans la partie du serveur qui serait actualisé pour chaque changement de connexion.

Lors du premier envoi d'un message d'un client, il y a un envoi vide à tous les autres utilisateurs. Ceci est dû à l'utilisation de la méthode *scanf* pour demander le « nickname ». On aurait pu utiliser *fgets*, or celle-ci ajoute un caractère de fin de ligne au string venant d'être lu. N'ayant pas eu assez de temps pour réparer ce défaut, je l'ai laissé dans ma version finale.

Conclusion

Pour conclure, on peut dire que dû à des fautes de ma part, ceci n'est pas un projet réussi à 100%. La fonctionnalité principale, le salon de discussion virtuel, fonctionne et on peut s'y échanger, mais ils manquent des fonctionnalités demandées dans l'énoncé que je n'ai pas implémenté.