



**INF3405 – Réseaux informatiques**

**Automne 2017**

**TP No. 4**

**Groupe 3**

**1793057 – Anthony Lachance**

**1795691 – Guillaume Larochelle**

**Soumis à : Aymen Djellal.**

**24 novembre 2017**

## Introduction

Dans le cadre du cours « Réseaux informatiques », nous avons dû faire une continuation du travail pratique trois pour nous familiariser avec l'utilisation des « Sockets ». Nous avons donc passé d'une communication unique entre un client et un serveur, pour ensuite faire une salle de clavardage « open source » pour plusieurs utilisateurs. Cependant, notre salle de clavardage ne se fera pas toute seule, nous avons un ami qui s'est occupé de la sécurité, un autre s'est occupé de l'esthétisme et nous nous sommes occupés de gérer les sockets/threads.

## Présentation du travail

Comme il a été mentionné plus haut, nous avons repris une partie du travail pratique trois pour avoir la base en ce qui concerne les sockets. Nous l'avons modifié pour que le serveur puisse gérer plusieurs connexions d'utilisateurs. Du côté serveur, nous avons commencé par lui demander le port d'écoute du serveur pour ensuite demander l'adresse IP du serveur. Ensuite, le serveur attend que des utilisateurs se connectent en envoyant leur nom d'utilisateur ainsi que leur mot de passe pour que le serveur vérifie dans sa base de données (un fichier texte) s'il y a une concordance. S'il y en a une, le serveur connecte l'utilisateur à la salle de clavardage sinon, l'utilisateur est ajouté à la base de données et ensuite envoyé vers la salle de clavardage. Nous avons créé une structure contenant les informations importantes des utilisateurs, ce qui inclut : le socket que l'utilisateur utilise, son adresse IP, le port qu'il utilise, son nom d'utilisateur ainsi que le contenu du message qu'il peut envoyer. Ces informations sont utiles parce qu'elles sont envoyées dans le thread de l'utilisateur. Dans le thread, le serveur peut recevoir les messages de tous les utilisateurs de la salle de clavardage pour ensuite les renvoyer vers tous les utilisateurs. Du côté client, le client commence par entrer l'adresse IP du serveur auquel il veut se connecter, le port d'écoute du serveur ainsi que son nom d'utilisateur et son mot de passe. Si toutes ces informations sont valides, elles seront envoyées vers le serveur pour son traitement. Après avoir été connecté à la salle de clavardage, le client peut envoyer et recevoir autant de message qu'il le désire, tant que ces messages ne dépassent pas 200 caractères. Si jamais l'utilisateur veut se déconnecter, il n'aura qu'à envoyer au serveur une chaîne vide.

## Difficultés rencontrées

Nous avons rencontré plusieurs difficultés lors de la rédaction de ce travail pratique, la première n'est pas vraiment une difficulté mais plutôt un irritant et qui est l'utilisation de `char*`. Même si l'utilisation d'un string est plus simple, la plupart des méthodes utilisées demandent l'utilisation de `char*` ce qui pose des ennuis de temps en temps. Un des problèmes a été avec l'envoi et la réception de message, après deux ou trois messages, l'utilisateur ne pouvait plus rien envoyer. Le problème était que nous avons oublié de vider le tableau de `char` après chaque envoi et chaque réception. Du côté serveur, nous avons dû créer une structure contenant les informations de l'utilisateur parce que nous ne pouvions pas passer par paramètre chaque information individuellement lors de la création d'un thread. Nous avons eu quelques ennuis avec le fichier de la base de données, nous avons eu la difficulté lorsque l'on ouvrait un fichier en lui disant que l'on pouvait écrire et lire dans ce fichier. Nous avons donc décidé de faire ces actions-là séparément, ce qui veut dire que l'on ouvre le fichier en écriture lorsque l'on veut que ça, nous avons fait la même chose lors de la lecture. Nous avons dû faire des deep copy du nom de l'utilisateur ainsi que son mot de passe lorsque nous créons un nouvel utilisateur parce que chaque nouvel utilisateur subséquent qui était créé écrasait ces informations des précédents utilisateurs.

## Critiques et Améliorations

Pour ce qui est de l'amélioration des laboratoires, il s'agit plutôt des cours théoriques que nous voudrions améliorer parce que la plupart des notions théoriques nécessaires pour les laboratoires venaient après les laboratoires en question. De plus nous n'avons reçu aucune notion utile dans le cours théorique pour la réalisation de ce travail pratique, nous avons dû faire des recherches de notre côté pour trouver des solutions à nos problèmes.

## **Conclusion**

Pour conclure, nous avons appris beaucoup de choses grâce à ce travail pratique. Nous avons entre autres appris à utiliser plus efficacement les sockets et les threads et à comprendre les difficultés qui y sont reliées. Nous devons aussi faire attention lors de la manipulation des sockets. Donc, nous pouvons dire que ce laboratoire nous a permis d'en apprendre plus sur les sockets grâce à l'auto-apprentissage.