



**POLYTECHNIQUE
MONTREAL**

**LE GÉNIE
EN PREMIÈRE CLASSE**

INF3405 – Réseaux informatiques

**Rapport du travail pratique #1
Gestionnaire de fichier**

Groupe 01

**Simon Nadeau – 1791314
Mathieu Châteauvert – 1846732**

**Travail soumis à
Mehdi Kadi**

**Rapport remis le 26 octobre 2018
École Polytechnique de Montréal**

Introduction

Le stockage de données sur des serveurs distants est de plus en plus accessible et utilisé de nos jours vu son côté pratique : il permet d'accéder à ses fichiers à partir de n'importe quelle machine. Toutefois, les grosses compagnies en profitent pour en retirer de l'argent en limitant la quantité de données qu'il est possible de sauvegarder sur leurs serveurs. Pour remédier à cette situation et éviter d'avoir à dépenser inutilement, ce laboratoire vise à développer une application client-serveur permettant de stocker des fichiers sur un serveur de stockage.

Plus précisément, le but de ce laboratoire est de développer une application réseau avec une simple interface console permettant de communiquer avec un serveur agissant à titre de gestionnaire de fichier. Les principaux objectifs sont donc de se familiariser avec le développement d'applications réseau en utilisant les threads et avec les échanges client-serveur en utilisant les sockets.

Présentation de la solution

Premièrement, notre solution comporte deux projets Eclipse : un projet pour le client et un projet pour le serveur. Pour exécuter les projets, il suffit de lancer les fichiers .jar sur les machines où l'on désire avoir les clients et le serveur. Plusieurs clients peuvent se connecter au même serveur, mais un client ne peut pas se connecter à plusieurs serveurs à la fois.

Le serveur doit d'abord être configuré avec une adresse IP et un port valides : les fonctions `validateIp()` et `validatePort()` de la classe `Server` s'occupent de s'assurer de la validité de ceux-ci. Par la suite, un `ServerSocket` est créé et se met en écoute de demandes de connexion provenant des clients. À ce moment, il est possible de configurer des clients en leur donnant l'adresse IP et le port du serveur. Ce sont les fonctions `initializeIp()` et `initializePort()` de la classe `Client` qui s'en occupent et qui vérifient la validité des informations fournies grâce aux méthodes `validateIp()` et `validatePort()` de la classe `Client`. Si tout est bon, un nouveau `Socket` est créé et une connexion est établie pour communiquer avec le serveur.

Lorsqu'une connexion est établie entre un client et le serveur, le serveur instancie un nouvel objet `Manager` dont la méthode principale `run()` est lancée sur un nouveau thread. Cette méthode consiste principalement en une boucle lisant (`readLine()`) et exécutant (`processCommand()`) les commandes provenant du client. Les commandes sont identifiées grâce à la méthode

`firstWordFromCommand()` et les paramètres (pour les commandes qui en ont) sont extraits grâce à la fonction `secondWordFromCommand()`. Par la suite, chaque commande est imprimée sur la console du serveur et est exécutée avec une implémentation qui lui est propre.

Du côté du client, après la création de la connexion avec le serveur, le seul travail effectué est la lecture des commandes entrées dans la console du client (`readLine()`), la validation de celles-ci (`isValidCommand()`) et l'impression de la réponse du serveur sur la console du client (`log()`) jusqu'à ce que la commande « exit » soit entrée. À ce moment, le Socket est fermé (`close()`), la connexion avec le serveur est rompue et le client est fermé.

Difficultés rencontrées

Le laboratoire s'est relativement bien déroulé : il n'y a pas eu de difficulté majeure. Les aspects qui auraient pu s'avérer plus complexes étaient la communication client-serveur et l'utilisation de threads, mais les exercices d'introduction au travail pratique #1 ont permis de bien se préparer et de comprendre comment utiliser ces concepts.

L'utilisation de la classe `File` a permis de facilement implémenter les commandes « ls » et « mkdir ». Toutefois, coder sur l'environnement Windows à l'école est un peu différent que de coder sur nos ordinateurs personnels. Par exemple, lorsqu'il est venu le temps d'implémenter sur les ordinateurs de l'école les chemins (*path*), nous avons rencontré des difficultés. Les fonctions ne fonctionnaient pas de la même façon et cela nous a causé des problèmes.

Par la suite, lorsqu'il est venu le temps d'implémenter le « download » et le « upload », la méthode que nous avons trouvée fermait le socket dès qu'on finissait de transférer le fichier. Nous avons dû passer arrêter la transmission dès que la fin du fichier était atteinte.

Critiques et améliorations

La difficulté de ce laboratoire ainsi que le temps consacré étaient raisonnables pour son poids dans la session. En effet, nous y avons consacré un total de 24 heures. Les exercices d'introduction ont été très utiles pour démarrer le travail et la documentation abondante sur java en ligne nous a permis de bien le poursuivre par la suite. De plus, l'énoncé du travail et les requis étaient clairs et précis.

Un point à améliorer serait peut-être au sujet de ce rapport. Celui-ci doit permettre l'évaluation de la qualité 7 du BCAPG concernant la communication. Toutefois, le contenu de celui-ci semble plutôt superflu. En effet, comme on nous demande de documenter notre code, la principale partie de ce rapport, soit la présentation de notre solution, est un peu redondante. Il faudrait repenser la méthode d'évaluation de la qualité 7 du BCAPG et demander un travail permettant d'avoir un rapport un peu plus complet et pertinent.

Autrement, le sujet du laboratoire et les objectifs d'apprentissage étaient très intéressants.

Conclusion

En conclusion, les objectifs de ce laboratoire étaient de se familiariser avec le développement d'applications réseau en utilisant les threads et avec les échanges client-serveur en utilisant les sockets. Pour ce faire, nous devions développer une application réseau avec une simple interface console permettant de communiquer avec un serveur agissant à titre de gestionnaire de fichier.

Ces objectifs ont bel et bien été atteints. En effet, grâce aux apprentissages effectués au cours de la réalisation du laboratoire, nous avons pu développer une application satisfaisant tous les requis. De plus, nous sommes maintenant plus à l'aise avec les concepts de socket et de thread et avec leur utilisation en java. Ces apprentissages seront probablement utiles pour la suite de notre parcours en informatique, puisque ce sont des concepts plutôt récurrents. Comme ils constituaient les principaux objectifs de ce laboratoire, nos attentes à l'égard de celui-ci ont été comblées.