POLYTECHNIQUE MONTRÉAL

UNIVERSITÉ D'INGÉNIERIE

Département de génie informatique et génie logiciel

LOG1410
Analyse et conception de logiciels

Laboratoire no. 1

Gabriel Bruyere (2248817)
Nils Coulier (2077378)
Olivier Tremblay-Noël (1903926)

Section de labo 03

Hiver 2023

Table des matières

1. Introduction

Dans le cadre de ce TP, nous avons réalisé une application client-serveur nous permettant de stocker des fichiers sur un serveur de stockage. Pour ce faire, nous avons implémenté une application console qui permet à l'utilisateur d'exécuter une variété de commandes pour interagir avec le serveur. Suite à la connexion au serveur, nous voulons avoir la possibilité de naviguer vers les répertoires enfant ou parent à partir du client, d'afficher la liste des fichiers dans le répertoire courant, de créer des dossiers vides, de téléverser des fichiers provenant du client vers le serveur, de télécharger des fichiers provenant du serveur vers notre répertoire local et enfin, lorsqu'on a fini, de se déconnecter du serveur de stockage. Enfin, nous voulons afficher en temps réél l'historique des commandes utilisées.

2. Présentation de vos travaux

Le projet a nécessité plusieurs bibliothèques pour l'implémentation de notre programme, soit : - java.io (pour afficher et lire sur le terminal) - java.net (pour connexion, communication et partage de données) - java.time (pour afficher le temps de chaque requête sur le server) - java.util (pour des manipulations de tableaux, pour l'utilisation de regex dans la vérification du format des adresses ip/numéros de port et enfin pour les Scanners qui nous permette de gérer l'interaction avec le client à travers la console)

La connexion et la communication entre le serveur et le client utilisent abondamment java.net. Tout d'abord, ils se connectent. Ensuite, les informations pour la connexion sont vérifiées du côté client (format du IP et intervalle du port), ceci utilise le java.io. Enfin, notre interface permet au client plusieurs tentatives avant de quitter le programme.

Le client a la possibilité d'envoyer des commandes au serveur, une à la fois. Les commandes sont des strings traitées selon leur contenu.

- cd permet de changer le dossier courant chez le serveur. Pour des raisons de sécurité, il est impossible de reculer plus bas que le dossier initial (celui où le server est lancé).
- ls retourne les dossiers et les fichiers contenus dans le dossier courant (triés par dossier, puis fichier, alphabétiquement).
- mkdir permet de créer un dossier s'il n'existe pas déjà dans le répertoire.
- upload et download permettent respectivement de téléverser et télécharger un dossier depuis le dossier spécifié et le copier dans le dossier courant (selon le cas).
- exit permet finalement au client de quitter le serveur et fermer la connexion.

3. Difficultés rencontrées lors de l'élaboration du TP et les éventuelles solutions apportées.

Nous avons rencontré quelques difficultés lors de la réalisation de cette application. Premièrement, puisque dans notre code nous avons fait des classes SendFile et ReceiveFile qui sont communes à l'implémentation du upload et du download nous avons eu des difficultés à garder le code réutilisable pour le client et le serveur.

Le client et le serveur étant agnostiques de leur situation mutuelle, il a été complexe de rendre la communication efficace pour terminer l'échange de fichier et de commande au bon moment selon toutes les circonstances du code (fichier inexistant, commande invalide, etc).

Implémenter un cd fonctionnel vérifiant l'existence et la validité du chemin spécifié, tout en s'assurant de ne pas pouvoir dépasser la racine virtuelle du serveur s'est révélé comme une difficulté qu'il a fallu surmonter afin de mener à bien notre projet. En effet, l'entrelacement des différents tableaux, et variables a nécessité une certaine précaution dans la rédaction du code.

Finalement, lier la notion de dossier courant (current directory) au sein des autres méthodes a été un défi.

4. Critiques et améliorations

Le TP est assez simple et il permet de se familiariser avec les échanges entre client et server, ainsi que les contraintes que cela incombe. Il permet d'explorer des connaissances connexes comme le regex, la gestion d'erreurs et le UX. C'est également l'occasion de se familiariser avec les bibliothèques mentionnées ci-haut.

Le TP est globalement bien balancé.

5. Conclusion

Nos attentes ont été comblées lors du TP. Notre grand-mère est contente d'avoir pu rendre son pentium 3 utile à la communauté. Cette expérience a été formatrice et nous a permis de comprendre ce qui se cache "sous le capot" des applications client-serveur ainsi que de comprendre les commmandes de base pour la navigation dans les terminaux (ls, cd, mkdir, etc). Finalement, le TP nous a sensibilisé la la transmission de fichiers sur un socket, qui doit absolument se faire en divisant un fichier volumineux par petits paquets.