



École Polytechnique de Montréal Département Génie Informatique et Génie Logiciel

INF3405 – Réseaux Informatiques

TP1 : Projet en réseaux informatiques Gestionnaire de fichier

1. Informations générales

Session	Automne 2018
Public cible	Étudiants de 1 ^{er} cycle
Date et Lieu de réalisation	À partir du 4 septembre 2018 au Laboratoire L-4708
Taille de l'équipe	2 étudiants
Pondération	15 %
Date de remise du projet	Remise pour tous les groupes le 26 octobre (23h55 au plus tard)
Directives particulières	<ul style="list-style-type: none">✓ Tout rapport sera pénalisé de 5 points s'il est soumis par une équipe dont la taille est différente de deux (02) étudiants sans l'approbation préalable du chargé de laboratoire.✓ Soumission par moodle uniquement (http://moodle.polymtl.ca).✓ <u>Soumission d'une archive compressée (.zip / .rar) contenant le rapport (en format PDF), les projets Eclipse contenant le client et le serveur et les exécutable .jar</u>✓ Toute soumission de l'archive en retard est pénalisée à raison de 5 points par heure de retard.
Chargé de laboratoire	Groupe 01 - Mehdi Kadi (mehdi.kadi@polymtl.ca) Groupe 02 - Bilal Itani (bilal.itani@polymtl.ca) Groupe 03/04 - Émilie Dion-Paquin (emilie.dion-paquin@polymtl.ca)
Auteurs :	Mehdi Kadi, Bilal Itani, Émilie Dion-Paquin

2. Connaissances requises

- Sockets
- Threads
- Programmation **Java uniquement**

3. Objectifs du laboratoire

L'objectif de ce laboratoire est de familiariser l'étudiant :

- aux échanges Client/Serveur en utilisant les sockets;
- au développement « d'applications réseau » en utilisant les threads.

Ce travail pratique consiste, par la même occasion, à évaluer deux des 12 qualités de l'ingénieur définies par le BCAPG (Bureau canadien d'agrément des programmes de génie). Le Bureau d'agrément a pour mandat d'attester que les futurs ingénieurs ont atteint ces 12 qualités à un niveau acceptable. Les deux qualités en question sont:

Qualité 4 (Conception) : capacité de concevoir des solutions à des problèmes d'ingénierie complexes et évolutifs et de concevoir des systèmes, des composants ou des processus qui répondent aux besoins spécifiés, tout en tenant compte des risques pour la santé et la sécurité publiques, des aspects législatifs et réglementaires, ainsi que des incidences économiques, environnementales, culturelle et sociales.

Qualité 7 (Communication) : habileté à communiquer efficacement des concepts d'ingénierie complexes, au sein de la profession et au public en général, notamment lire, rédiger, parler et écouter, comprendre et rédiger de façon efficace des rapports et de la documentation pour la conception, ainsi qu'énoncer des directives claires et y donner suite.

4. Description

Mise en contexte

Vous en avez marre de ne plus avoir d'espace de stockage sur votre compte Dropbox ou Google Drive. À chaque fois que vous essayez de mettre un fichier sur le nuage, les géants de l'infonuagique essaient de vous soutirer de l'argent de vos petites poches pour vous vendre encore plus de stockage. La vie d'étudiant étant rude, et sachant que vous n'avez pas d'argent à donner à ces compagnies, vous choisissez de vous révolter. Étant étudiant à Polytechnique Montréal et surtout, expert en réseau informatique avant même d'avoir fini le cours, vous choisissez de développer votre propre application client-serveur permettant de stocker n'importe quel type de fichier sur un serveur de stockage. Votre grand-mère, une fidèle supporter, vous encourage dans votre quête et vous fait don de son bon vieux Pentium 3. Vous choisissez de l'utiliser comme serveur de stockage.

Pour l'instant, vous prévoyez qu'une simple **interface console**, car vous voulez assurer uniquement le fonctionnement de votre application client-serveur et une interface graphique n'est pas une de vos priorités.

Requis :

Vous aurez à faire deux produits, un serveur, et un client pour communiquer avec le serveur.

Au démarrage du serveur, celui-ci demande à l'utilisateur d'entrer les informations suivantes : adresse IP du poste sur lequel s'exécute le serveur, port d'écoute (**un port entre 5000 et 5050 uniquement**). Une vérification doit être faite pour s'assurer que l'utilisateur a bien entré des données cohérentes. Par exemple, le serveur doit détecter que l'adresse 7x7.-202.666.888 et le port -9999 sont incohérents. Il devra aussi s'assurer que l'adresse IP entrée est bien sur quatre octets.

Au lancement du client, celui-ci demande à l'utilisateur d'entrer l'adresse IP du serveur, le port du serveur (**entre 5000 et 5050**). Le client doit vérifier la validité du format de l'adresse IP et du port entré par l'utilisateur (de la même façon que le fait le serveur) et afficher une erreur en cas d'invalidité. Ces informations sont utilisées pour tenter de se connecter au serveur. Une fois la connexion établie entre le client et le serveur de stockage, le client peut, **à tout moment** lancer une des cinq commande suivante :

Commande	Description
cd <Nom d'un répertoire sur le serveur> <i>Note : Vous devez aussi implémenter le '..' pour se déplacer vers un répertoire parent</i>	Commande permettant à l'utilisateur de se déplacer vers un répertoire enfant ou parent
ls	Commande permettant d'afficher à l'utilisateur tous les dossiers et fichiers dans le répertoire courant de l'utilisateur au niveau du serveur.
mkdir <Nom du nouveau dossier>	Commande permettant la création d'un dossier au niveau du serveur de stockage.
upload <Nom du fichier>	Commande permettant le téléversement d'un fichier, se trouvant dans le répertoire locale du client, vers le serveur de stockage
download <Nom du fichier>	Commande permettant le téléchargement d'un fichier, se trouvant dans le répertoire courant de l'utilisateur au niveau du serveur de stockage, vers le répertoire local du client.
exit	Commande permettant au client de se déconnecter du serveur de stockage.

Quand le serveur reçoit une commande la part d'un client, il doit afficher dans sa console les informations suivantes : [Adresse IP client : Port client - Date et Heure (min, sec)] : <Commande> par exemple :

```
[132.207.29.107:42975 - 2018-09-15@13:02:01] : upload allo.docx
```

```
[132.207.29.107:42975 - 2018-09-15@13:02:05] : ls
```

```
[132.207.29.122:35928 - 2018-09-15@13:02:08] : download NotesDeCoursINF3405.pdf
```

L'exemple ci-haut montre aussi que votre serveur devra être capable de supporter la connexion de plusieurs clients à la fois, il faudra utiliser des *threads* au niveau du serveur pour réussir à supporter ces différents clients.

Veuillez consulter l'annexe de ce document pour un exemple d'affichage attendu lors du lancement des différentes commandes.

Notez que l'application client et l'application serveur ne sont (en théorie) pas exécutées sur la même machine et que, bien évidemment, plusieurs clients peuvent tenter de se connecter au serveur en même temps. Cependant, chaque client connaît d'avance l'adresse IP du serveur ainsi que le numéro de port écouté.

Afin de pouvoir valider le bon envoi et la bonne réception de vos fichiers, vous pouvez vous servir de l'outil `certUtil` disponible dans une console Windows. Lors de la correction du laboratoire, nous effectuerons aussi nos vérification à l'aide de cet outil. Pour vérifier le hash d'un fichier, il suffit de lancer la commande comme suit dans un terminal Windows:

```
certUtil -hashfile <chemin/vers/le/fichier> <algorithme de hashage>
```

Nous utiliserons l'algorithme de hashage MD5 pour effectuer nos tests sur l'intégrité d'envoi d'un fichier. L'idée est de s'assurer que le hash du fichier au niveau du client correspond au hash du fichier au niveau du serveur de stockage. Le cas échéant, l'envoi du fichier vers le serveur a été effectué avec succès.

Votre client et votre serveur doivent être bien développés, comme si vous alliez le livrer à un de vos clients. Une erreur d'exécution entraînera une pénalité.

5. Résumé des requis fonctionnels

Qualité évaluée :

4.3 Procéder à la conception

Critère d'évaluation : Intégrer les concepts de programmation en réseautique retenus au premier laboratoire en répondant aux besoins et en respectant les requis fonctionnels du projet courant.

En somme, les requis fonctionnels sont :

1. Saisie des paramètres du serveur (adresse IP, port d'écoute entre 5000 et 5050)
2. Vérifier la validité de l'adresse IP saisie (uniquement le format) et le numéro de port (entre 5000 et 5050)
3. Téléverser et télécharger un fichier
4. Pouvoir se déplacer dans la hiérarchie des répertoires du serveur de stockage à partir du client
5. Pouvoir énumérer les répertoires et les fichiers au niveau du serveur de stockage à partir du client
6. Pouvoir créer un répertoire à partir du client sur le serveur de stockage à partir du client
7. Pouvoir se déconnecter adéquatement du serveur de stockage
8. Afficher en temps réel les demandes à traiter (logs au niveau de la console serveur)

6. Livrable

1. Langages et bibliothèques autorisés

- Le client et le serveur doivent être développés uniquement en Java.
- Usage permis de librairies externes comme JSON.simple ou GSON. Si vous utilisez ces librairies, veuillez justifier dans la section « Présentation » de votre rapport la raison de leur utilisation.

Tout non-respect de ces consignes entraînera la note de 0.

2. Soumission

Le livrable est une archive (ZIP ou RAR) ayant le format suivant:

INF3405_TP1_matriculeX_matriculeY où matriculeX > matriculeY

Votre archive contiendra les fichiers suivants :

- Les projets Eclipse du client et du serveur incluant les fichiers sources (.java), autrement dit le dossier en entier contenant votre projet
- Le rapport au format PDF
- Les fichiers exécutables de votre client et votre serveur (.jar)

****Assurez-vous que les livrables compilent et s'exécutent adéquatement sur les ordinateurs du laboratoire ! ****

3. Rapport

Qualité évaluée :

7.1 Lire et rédiger de la documentation

Critère d'évaluation : Rédiger un rapport technique documentant efficacement le travail d'ingénierie réalisé dans ce projet en utilisant différentes formes de langage (naturel, informatique, etc.)

Le rapport, d'une longueur maximale de 3 pages (excluant la page de présentation), doit comporter les éléments suivants :

- **Page présentation** qui doit contenir le nom ou le logo de l'école, le libellé et l'identifiant du cours, la session, le numéro et l'identification du projet, la date de remise, les matricules et noms des membres de l'équipe, la mention « Soumis à : **nom et prénom du chargé de laboratoire** ».
- **Introduction** en vos propres mots pour mettre en évidence le contexte et les objectifs du TP.

- **Présentation** de vos travaux. Une explication de votre solution mettant en lumière la prise en compte des principaux requis du système. Si vous utilisez des configurations particulières des bibliothèques ou des projets, précisez-les également.
- **Difficultés rencontrées** lors de l'élaboration du TP et les éventuelles solutions apportées.
- **Critiques et Améliorations** : Il serait intéressant d'inclure vos suggestions pour améliorer le laboratoire.
- **Conclusion** : Expliquez en quoi ce laboratoire vous a été utile, ce que vous avez appris, si vos attentes ont été comblées, etc.

7. Évaluation

Évaluation de l'exécutable	6
Évaluation de l'implémentation : gestion adéquate des variables et de toute ressource (création, utilisation, libération), gestion des erreurs, logique de développement, documentation du code , etc.	8
Rapport	6
Total des points	20

8. Annexe

Exemples d’affichage au niveau du client des diverses commandes:

Commande ls

ls

[Folder] INF3405

[Folder] bonjour

[File] INF3405.pdf

Commande mkdir

mkdir bonsoir

Le dossier bonsoir a été créé.

Commande cd

cd bonsoir

Vous êtes dans le dossier bonsoir.

Commande upload

upload bye.jpg

Le fichier bye.jpg à bien été téléversé.

Commande download

download bye.jpg

Le fichier bye.jpg à bien été téléchargé

Commande exit

exit

Vous avez été déconnecté avec succès.