
Projet : RMI & JDBC

Modalités :

- ce projet est un projet individuel : au plus, une personne par groupe !
- date limite : vendredi 28 novembre.
- pas de soutenance.
- la note de contrôle continu sera égale à la note de projet.
- toute fraude détectée, notamment copié/collé, sera sanctionnée par une note ≤ 5 : pensez à interdire le droit d'accès en lecture de vos documents... (sous unix, `chmod og-r monProjet`)
- les noms de vos classes devront respecter la syntaxe suivante : `votrenomClasse.java`

Livrables :

- les codes sources, abondamment documentées, et archivées : `votrenomPOA.tar` ou `votrenom-POA.jar` selon le format d'archivage choisi.
- un rapport d'au plus quatre pages expliquant l'architecture retenue, et justifiant les choix techniques, en fonction des difficultés rencontrées.

Description :

Le but du projet est d'implémenter un serveur de fichiers distribué, analogue à ceux des réseaux pair-à-pair : chaque fichier téléchargeable est fragmenté et distribué sur les machines des utilisateurs enregistrés. Pour télécharger un fichier, on récupère donc un ensemble de fragments auprès de multiples machines, permettant de reconstituer le fichier dans son intégralité.

L'intérêt de cette architecture décentralisée est de permettre le transfert simultané des fragments et la tolérance aux pannes (si un ensemble de dépôts devient inaccessible, le téléchargement des fragments concernés est repris à partir d'un autre ensemble de dépôts les hébergeant).

Contraintes :

- l'ensemble des données concernant les utilisateurs du réseau, les dépôts, les fichiers hébergés, la localisation des fragments sera confié à une base de données.
- votre projet doit contenir au minimum 3 classes : `votrenomClientBDD.java`, `votrenomServeurFichiers.java` et `votrenomClient.java`
 - `votrenomClientBDD.java` : classe java dédiée à l'ensemble des opérations affectant la base de donnée (création, interrogation, mise à jour)
 - `votrenomServeurFichiers.java` : classe java gérant le téléchargement des multiples fragments, entre autres.
 - `votrenomClient.java` : classe java prenant en charge l'interface utilisateur - elle accèdera à une instance de `votrenomServeurFichiers`, via RMI.
- fonctionnalités requises :
 - barre de progression du téléchargement du fichier demandé
 - tolérance aux pannes simple : si un dépôt devient inaccessible, alors le téléchargement doit se poursuivre normalement.
- extras :
 - interface utilisateur conviviale, pour le choix du serveur, le choix du fichier à télécharger (download), le choix du fichier à mettre à disposition (upload), le choix du répertoire pour le stockage de fragments (dépôt) ...
 - téléchargements simultanés des fragments (threads)
 - tolérance aux pannes multiples (dépôts)
 - tolérance aux pannes du serveur de fichier
 - ...

La notation portera principalement sur la simplicité de l'architecture retenue, et sur l'élégance de l'implémentation.