

Mini-TP 1 : Serveur multithread des fichiers

On souhaite écrire un serveur des fichiers multithread qui permet de télécharger un fichier stocké sur le serveur et envoyer son contenu au client. Pour cela, il faut écrire une classe `FileServer` dont les méthodes principales sont :

1. `initWriterBuffer` et `initReaderBuffer`
2. `uploadFile` qui prend en paramètre un buffer d'écriture et le nom d'un fichier. Cette méthode transfère le fichier du client via le buffer d'écriture. Le fichier sera stocké sur le serveur.
3. `downloadFile` qui prend en paramètre un buffer de lecture et le nom d'un fichier. Cette méthode récupère le fichier d'un client via le buffer de lecture.
4. un constructeur qui reçoit la demande de connexion de clients, crée une socket de communication et crée un thread "fileThread" qui s'occupe de la vérification/l'échange de fichiers avec un client.
5. Afin de demander ce service au serveur, un client doit envoyer un objet d'une classe `Query` dont les attributs sont les suivants :

```
class Query
{
    int code; // 1 : upload; 2: dowload
    String fileName;
    autre... // a compléter en cas de besoin
}
```

6. Dans tous les cas, le serveur doit renvoyer un objet d'une classe `Response` dont les attributs sont les suivants :

```
class Response
{
    int code; //0 : accepté; 1 : refusé
    ...
}
```

7. Créez une interface avec JavaFX permettant au client d'échanger graphiquement avec le serveur. Les réponses du serveur doivent s'afficher sur l'interface.

Ecrire une classe `MainServer` dont la fonction "main" prend en paramètre (positionnel) un port. Cette fonction main crée un objet de la classe `FileServer`.

Mini-TP 2 : Serveur multithread des fichiers

1. Ecrivez le client Echo en java et testez le avec le serveur Echo en C. Le serveur doit renvoyer la même chaîne mais inversée (effet miroir).
2. Ecrivez une interface graphique en javaFX permettant au client de saisir une phrase et d'afficher le retour du serveur. L'interface sera décomposée en deux zones : zone de saisie côté client et zone serveur, où vous afficherez les réponses du serveur.
3. Ajouter une zone d'affichage des Log : code retour de toutes les fonctions !
4. Ajouter une fonction côté serveur qui stocke toutes les chaînes reçues par un client, les sauvent dans un tableau de chaînes de caractères
5. Ajouter une fonction côté client qui demande la liste de toutes les chaînes envoyées et les affiche à la console.