

## Projet Réseaux : Gestion de comptes bancaires

L'objectif est d'implémenter une application permettant de gérer des comptes bancaires en utilisant une architecture de type client-serveur.

Le serveur maintiendra pour chaque client un ou plusieurs comptes. On supposera que la création des comptes a été déjà faite.

Le serveur devra implanter les requêtes suivantes :

- **AJOUT** <id\_client id\_compte password somme> permettra d'ajouter la **somme** sur le compte identifié par **id\_compte** par le client **id\_client** identifié avec **password**.
- **RETRAIT** <id\_client id\_compte password somme> permettra de retirer la **somme** sur le compte identifié par **id\_compte** par le client **id\_client** identifié avec **password**.
- **SOLDE** <id\_client id\_compte password> permettra d'obtenir le solde du compte identifié par **id\_compte** par le client **id\_client** identifié avec **password**.
- **OPERATIONS** <id\_client id\_compte password> permettra d'obtenir les 10 dernières opérations effectuées sur le compte identifié par **id\_compte**. Cette opération ne peut être demandée que par le client **id\_client** identifié avec **password**.

Le client recevra et traitera les réponses du serveur :

- **OK** – informe le client que la commande (AJOUT, RETRAIT) s'est bien déroulée
- **KO** - informe le client que la commande ne s'est pas bien déroulée
- **RES\_SOLDE** - *solde date\_dernière\_opération* : envoi au client le solde actuel du compte et la date de la dernière opération.
- **RES\_OPERATIONS** - envoi au client les 10 dernières opérations du compte sous le format *type\_opération date\_opération montant\_opération*

Les documents :

1. <https://developer.ibm.com/tutorials/l-sock/>
2. <https://www.csd.uoc.gr/~hy556/material/tutorials/cs556-3rd-tutorial.pdf>
3. <https://broux.developpez.com/articles/c/sockets/#LV>
4. [https://www.bogotobogo.com/cplusplus/sockets\\_server\\_client.php](https://www.bogotobogo.com/cplusplus/sockets_server_client.php)

vous propose la présentation des sockets et leur utilisation dans différentes applications de type client-serveur (Windows et Linux). En vous inspirant du code fourni vous devez proposer :

- une implantation sous Linux du serveur et du client en utilisant des sockets TCP.
- une implantation sous Linux du serveur et du client en utilisant des sockets UDP.
- (bonus) Modifier votre serveur afin d'accepter et traiter plusieurs clients.

Travail à rendre via le moodle:

1. le code .c du serveur et du client.

2. Le makefile
3. le fichier Readme.txt permettant à l'utilisateur la compilation et l'exécution de l'application
4. un rapport présentant la description du code ainsi que le résultat des tests effectués (copies d'écran).