

Rapport de Projet - Gestion de Comptes Bancaires

1. Introduction

Ce projet a pour objectif de développer une application de gestion de comptes bancaires permettant d'effectuer des opérations de base telles que la consultation de solde, l'ajout et le retrait d'argent, ainsi que la consultation de l'historique des transactions. L'application utilise des serveurs TCP et UDP pour gérer les communications entre les clients et le serveur. Afin de garantir la gestion simultanée de plusieurs clients, les serveurs utilisent des threads.

2. Description du Code

2.1 Serveur TCP

Le serveur TCP utilise des threads pour gérer plusieurs clients simultanément. Chaque connexion client est gérée dans un thread dédié, assurant ainsi que les opérations des différents clients ne se bloquent pas mutuellement.

Fonctionnalités :

- **SOLDE** : Consultation du solde du compte.
- **AJOUT** : Ajout d'argent sur le compte.
- **RETRAIT** : Retrait d'argent du compte (vérification du solde disponible).
- **OPERATIONS** : Affichage des 10 dernières opérations.

2.2 Serveur UDP

Le serveur UDP gère également plusieurs clients grâce aux threads. Chaque message reçu est traité dans un thread indépendant, ce qui permet de traiter plusieurs requêtes simultanément sans blocage.

Fonctionnalités :

- Identiques à celles du serveur TCP : SOLDE, AJOUT, RETRAIT, OPERATIONS.

3. Procédure de Test

Pour les détails de la procédure de test, veuillez consulter le fichier **README.md**.

4. Résultats des Tests

1. Test de la commande AJOUT

Objectif : Ajouter de l'argent à un compte.

Test réussi :

AJOUT 001 001 pass12345 500

Résultat attendu : OK

Mot de passe incorrect :

AJOUT 001 001 wrongpass 500

Résultat attendu : KO

Compte inexistant :

AJOUT 003 003 pass12345 500

Résultat attendu : KO

2. Test de la commande RETRAIT

Objectif : Retirer de l'argent d'un compte.

Test réussi :

RETRAIT 001 001 pass12345 200

Résultat attendu : OK

Fonds insuffisants :

RETRAIT 001 001 pass12345 10000

Résultat attendu : KO

Mot de passe incorrect :

RETRAIT 001 001 wrongpass 200

Résultat attendu : KO

Yedam KIM
EI4
Projet Réseaux
15.01.2025

Capture D'écran:

TCP:

```
> AJOUT 001 001 pass12345 500
✓ Succès : OK

> AJOUT 001 001 wrongpass 500
✗ Échec : KO

> AJOUT 003 003 pass12345 500
✗ Échec : KO

> RETRAIT 001 001 pass12345 200
✓ Succès : OK

> RETRAIT 001 001 pass12345 10000
✗ Échec : KO

> RETRAIT 001 001 wrongpass 200
✗ Échec : KO

> 
```

Yedam KIM
EI4
Projet Réseaux
15.01.2025

UDP:

```
yedam@yedam-xps13p:~/Desktop/Projet_Reseaux$ ./clie_udp
Connecté au serveur UDP. Entrez vos commandes :
Exemples :
  AJOUT <id_client> <id_compte> <password> <montant>
  RETRAIT <id_client> <id_compte> <password> <montant>
  SOLDE <id_client> <id_compte> <password>
  OPERATIONS <id_client> <id_compte> <password>
  EXIT pour quitter.
> AJOUT 001 001 pass12345 500
✓ Succès : OK

> AJOUT 001 001 wrongpass 500
✗ Échec : KO

> AJOUT 003 003 pass12345 500
✗ Échec : KO

> RETRAIT 001 001 pass12345 200
✓ Succès : OK

> RETRAIT 001 001 pass12345 10000
✗ Échec : KO

> 
```

Yedam KIM
EI4
Projet Réseaux
15.01.2025

3. Test de la commande SOLDE

Objectif : Consulter le solde actuel d'un compte.

Test réussi :

SOLDE 001 001 pass12345

Résultat attendu : RES_SOLDE <montant> <date_derniere_operation>

Mot de passe incorrect :

SOLDE 001 001 wrongpass

Résultat attendu : KO

Compte inexistant :

SOLDE 003 003 pass12345

Résultat attendu : KO

4. Test de la commande OPERATIONS

Objectif : Afficher les 10 dernières opérations du compte.

Test réussi :

OPERATIONS 001 001 pass12345

Résultat attendu :

RES_OPERATIONS

AJOUT 2024-01-10 14:35:00 500.00

RETRAIT 2024-01-10 15:00:00 200.00

Mot de passe incorrect :

OPERATIONS 001 001 wrongpass

Résultat attendu : KO

Compte inexistant :

OPERATIONS 003 003 pass12345

Résultat attendu : KO

Yedam KIM
EI4
Projet Réseaux
15.01.2025

Capture D'écran:

TCP:

```
> SOLDE 001 001 pass12345
➡ Résultat : RES_SOLDE 0.00 2025-01-10 17:00:06

> SOLDE 001 001 wrongpass
❌ Échec : KO

> SOLDE 003 003 pass12345
❌ Échec : KO

> OPERATIONS 001 001 pass12345
📋 Historique des opérations :
RES_OPERATIONS

> AJOUT 001 001 pass12345 500
✅ Succès : OK

> RETRAIT 001 001 pass12345 200
✅ Succès : OK

> OPERATIONS 001 001 pass12345
📋 Historique des opérations :
RES_OPERATIONS
AJOUT 2025-01-10 17:00:39 500.00
RETRAIT 2025-01-10 17:00:50 200.00
```

Yedam KIM
EI4
Projet Réseaux
15.01.2025

UDP:

```
> SOLDE 001 001 pass12345
➡ Résultat : RES_SOLDE 300.00 2025-01-10 17:09:53

> SOLDE 001 001 wrongpass
❌ Échec : KO

> SOLDE 003 003 pass12345
❌ Échec : KO

> OPERATIONS 001 001 pass12345
📋 Historique des opérations :
RES_OPERATIONS
AJOUT 2025-01-10 17:09:17 500.00
RETRAIT 2025-01-10 17:09:28 200.00

> OPERATIONS 001 001 wrongpass
❌ Échec : KO

> OPERATIONS 003 003 pass12345
❌ Échec : KO

>
```

5. Conclusion et Bonus

Le projet de gestion de comptes bancaires via les serveurs TCP et UDP fonctionne correctement. L'utilisation des **threads** permet de gérer plusieurs clients simultanément sans blocage. Les tests effectués confirment le bon fonctionnement des différentes fonctionnalités. Le système est stable et robuste, répondant efficacement aux commandes des clients.