



POLYTECH®  
SORBONNE



SORBONNE  
UNIVERSITÉ

# Réseaux et communications

## Projet : gestion de comptes bancaires

## Commentaire du code :

Tous les fichiers ont déjà été allégrement commentés.

## **Fonctionnement des fichiers server.c :**

1 – La structure d'un client est déclarée et contient un identifiant, un mot de passe, un tableau de numéros de compte (un client peut avoir jusqu'à 10 comptes), un solde pour chaque compte ainsi que les 10 dernières opérations de chaque compte. [TCP et UDP]

2 – On rentre dans le main, tout se fera dans ce dernier. [TCP et UDP]

3 – On déclare d'abord toutes les variables dont on aura besoin plus tard. On initialise ainsi 4 clients à l'aide de la structure précédemment décrite et on les place dans un tableau. [TCP et UDP]

4 - On crée un socket serveur puis on configure l'adresse, le port et le type d'adresses utilisables à l'aide d'une structure sockaddr\_in. L'adresse IP est 127.0.0.1 et le port est aléatoire. [TCP et UDP]

5 – On lie le socket serveur à l'adresse spécifiée à l'aide de la fonction bind [TCP et UDP]

6 – le serveur est en attente de connexion avec une file d'attente créée avec la fonction listen. [TCP]

7 – Une fois qu'un client se connecte, on utilise la fonction accept pour établir proprement la connexion. [TCP]

8 - Un message est affiché pour indiquer que le serveur a accepté la connexion du client. [TCP]

## **A partir d'ici, TCP et UDP sont quasiment identiques**

9 – Le serveur envoie des demandes d'identifiant, de mot de passe et de numéro de compte au client et récupère les données reçues une par une.

10 – Le serveur vérifie si il y a correspondance entre l'identifiant reçu et les identifiants des clients déjà connus.

11.a – Si il n’y a pas correspondance, on rentre dans une boucle et on redemande l’identifiant tout en refaisant le test de correspondance fait précédemment jusqu’à ce que l’identifiant entré soit correct. Lors de la correspondance, on fait passer la variable `indic_verif` à 1 (la variable qui nous faisait rester dans la boucle va maintenant pouvoir nous en faire sortir) et on retient aussi l’index du client dans le tableau des clients.

11.b – Si il y a correspondance directement , on retient l’index du client dans le tableau des clients et on passe la variable `indic_verif` à 1, ce qui empêchera de rentrer dans la boucle.

12 – On répète le même processus pour le mot de passe et le numéro de compte reçu.

13 – On rentre dans la boucle des opérations dont on ne peut sortir uniquement si la variable `sortie_while` passe à 1.

14 – On envoie au client les différentes opérations possibles et on reçoit son choix d’opération.

15 – A l’aide d’un switch case, on traite les différents cas et opérations possibles.

16.a – Si 1 est reçu, on procède à un dépôt d’argent. On demande au client le montant qu’il veut retirer, puis après avoir ajouté ce montant au solde de son compte, on ajoute cette opération à la liste des 10 dernières opérations.

16.b – Si 2 est reçu, on fait le même processus mais pour un retrait d’argent.

16.c – Si 3 est reçu, on envoie la solde du compte du client au client.

16.d – Si 4 est reçu, on envoie les chaînes de caractères des 10 dernières opérations réalisées.

16.e – Si 5 est reçu, on passe la variable `sortie_while` à 1 ce qui nous fera sortir de la boucle while et terminer le programme.

16.f – Si autre chose est reçu, on indique que le choix est invalide et on repart au début de la boucle.

17 – Une fois sorti de la boucle, on ferme les sockets et le programme est terminé.

## **Fonctionnement des fichiers client.c :**

1 – On rentre dans le main, tout se fera dans ce dernier. [TCP et UDP]

2 – On déclare d'abord toutes les variables dont on aura besoin plus tard. [TCP et UDP]

3 - On crée un socket client puis on configure l'adresse, le port et le type d'adresses utilisables à l'aide d'une structure sockaddr\_in. L'adresse IP est 127.0.0.1 et le port est aléatoire. [TCP et UDP]

4 – On se connecte au serveur en utilisant la fonction connect. [TCP]

### **A partir d'ici, TCP et UDP sont quasiment identiques**

5 – Le client reçoit les demandes d'identifiant, de mot de passe et de numéro de compte du client et envoie toutes ces informations.

6 – Le serveur vérifie si il y a correspondance entre l'identifiant reçu et les identifiants des clients déjà connus et le client reçoit de ce dernier des feu vert ou non.

7.a – Si il n'y a pas correspondance, on rentre dans une boucle et on renvoie l'identifiant jusqu'à ce qu'on reçoive le feu vert du serveur au niveau de la correspondance.

7.b – Si il y a correspondance directement , on saute la boucle car le feu vert a directement été envoyé.

8 – On répète le même processus pour le mot de passe et le numéro de compte reçu.

13 – On rentre dans la boucle des opérations dont on ne peut sortir uniquement si la variable sortie\_while passe à 1.

14 – On reçoit du serveur les différentes opérations possibles et on envoie son choix d'opération.

15 – A l'aide d'un switch case, on traite les différents cas et opérations possibles.

16.a – Si 1 est envoyé, on procède à un dépôt d'argent. On reçoit la demande du montant qu'on veut retirer, puis après on envoie ce montant (somme limite, si on envoie un mauvais montant, on doit réécrire le montant).

16.b – Si 2 est reçu, on fait le même processus mais pour un retrait d'argent.

16.c – Si 3 est reçu, on reçoit la solde du compte du client et on l'affiche.

16.d – Si 4 est reçu, on reçoit et affiche les 10 dernières opérations réalisées.

16.e – Si 5 est reçu, on passe la variable sortie\_while à 1 ce qui nous fera sortir de la boucle while et terminer le programme.

16.f – Si autre chose est reçu, on lit que le choix est invalide et on repart au début de la boucle.

17 – Une fois sorti de la boucle, on ferme les sockets et le programme est terminé.

## **Captures d'écran :**

Connexion au compte (cas où il n'y a pas d'erreurs d'identifiant, de mot de passe ainsi que de numéro de compte) :

### **Client**

```
paulbouchard@air-de-paul TCP % ./client
-----
Bienvenue à la banque, veuillez vous connecter avant de faire une opération.
Entrez votre identifiant : Paul
Entrez votre mot de passe : AH
Entrez votre numéro de compte : 111
Identifiant validé
Mot de passe validé
Numéro de compte validé
```

### **Server**

```
paulbouchard@air-de-paul TCP % ./server
server: got connection from 127.0.0.1 port 56307
-----
Interface de la banque
Identifiant du client : Paul
Identifiant validé
Mot de passe du client : AH
Mot de passe validé
Numéro de compte du client : 111
Numéro de compte validé
```

Connexion au compte (cas où il y a une erreur d'identifiant et de numéro de compte) :

## Client

```
[paulbouchard@air-de-paul TCP % ./client
-----
Bienvenue à la banque, veuillez vous connecter avant de faire une opération.
Entrez votre identifiant : polo
Entrez votre mot de passe : AH
Entrez votre numéro de compte : bug
Votre identifiant est invalide. Veuillez le retaper : chaima
Votre identifiant est invalide. Veuillez le retaper : Paul
Identifiant validé
Mot de passe validé
Votre numéro de compte est invalide. Veuillez le retaper : rebug
Votre numéro de compte est invalide. Veuillez le retaper : debug
Votre numéro de compte est invalide. Veuillez le retaper : 1111
Votre numéro de compte est invalide. Veuillez le retaper : 111
Numéro de compte validé
```

## Server

```
[paulbouchard@air-de-paul TCP % ./server
server: got connection from 127.0.0.1 port 56319
-----
Interface de la banque
Identifiant invalide
Identifiant invalide
Identifiant du client : Paul
Identifiant validé
Mot de passe du client : AH
Mot de passe validé
Numéro de compte invalide
Numéro de compte invalide
Numéro de compte invalide
Numéro de compte invalide
Numéro de compte du client : 111
Numéro de compte validé
```

Affichage des dernières opérations effectuées puis crédit sur le compte puis ré-affichage des dernières opérations réalisées :

## Client

```
-----
Tapez le numéro en face d'une manoeuvre pour exécuter cette dernière
1 - Créditer votre compte
2 - Débiter votre compte
3 - Afficher la solde de votre compte
4 - Afficher les 10 dernières opérations effectuées
5 - Quitter la banque

4

12/03/2023 : débit de 200€
24/04/2023 : crédit de 1000€
31/05/2023 : débit de 1300€

Tapez le numéro en face d'une manoeuvre pour exécuter cette dernière
1 - Créditer votre compte
2 - Débiter votre compte
3 - Afficher la solde de votre compte
4 - Afficher les 10 dernières opérations effectuées
5 - Quitter la banque

1

Entrez la somme que voulez créditer à votre compte (entre 1€ et 1000000€) : 3000
OK

Tapez le numéro en face d'une manoeuvre pour exécuter cette dernière
1 - Créditer votre compte
2 - Débiter votre compte
3 - Afficher la solde de votre compte
4 - Afficher les 10 dernières opérations effectuées
5 - Quitter la banque

4

23/12/2023 : crédit de 3000 €
12/03/2023 : débit de 200€
24/04/2023 : crédit de 1000€
31/05/2023 : débit de 1300€
```

## Server

```
-----
Compte de Paul
=====
Affichage des 10 dernières opérations
=====
Ajout de 3000 € à la solde du compte
Nouvelle solde : 5000 €
=====
Affichage des 10 dernières opérations
□
```

Affichage de la solde, débit puis crédit puis ré-affichage du solde :

## Client

Votre solde actuelle est de 5000 €

Tapez le numéro en face d'une manoeuvre pour exécuter cette dernière

- 1 - Créditer votre compte
- 2 - Débiter votre compte
- 3 - Afficher la solde de votre compte
- 4 - Afficher les 10 dernières opérations effectuées
- 5 - Quitter la banque

2

Entrez la somme que voulez débiter de votre compte (entre 1€ et 10000€) : 4500  
OK

Tapez le numéro en face d'une manoeuvre pour exécuter cette dernière

- 1 - Créditer votre compte
- 2 - Débiter votre compte
- 3 - Afficher la solde de votre compte
- 4 - Afficher les 10 dernières opérations effectuées
- 5 - Quitter la banque

1

Entrez la somme que voulez créditer à votre compte (entre 1€ et 1000000€) : 23  
OK

Tapez le numéro en face d'une manoeuvre pour exécuter cette dernière

- 1 - Créditer votre compte
- 2 - Débiter votre compte
- 3 - Afficher la solde de votre compte
- 4 - Afficher les 10 dernières opérations effectuées
- 5 - Quitter la banque

3

Votre solde actuelle est de 523 €

## Server

```
Affichage du solde du compte
=====
Débit de 4500 € de la solde du compte
Nouvelle solde : 500 €
=====
Ajout de 23 € à la solde du compte
Nouvelle solde : 523 €
=====
Affichage du solde du compte
```



Affichage des dernières opérations puis déconnexion de la banque :

## Client

```
Tapez le numéro en face d'une manoeuvre pour exécuter cette dernière
1 - Créditer votre compte
2 - Débiter votre compte
3 - Afficher la solde de votre compte
4 - Afficher les 10 dernières opérations effectuées
5 - Quitter la banque

4

23/12/2023 : crédit de 23 €
23/12/2023 : débit de 4500 €
23/12/2023 : crédit de 3000 €
12/03/2023 : débit de 200€
24/04/2023 : crédit de 1000€
31/05/2023 : débit de 1300€

Tapez le numéro en face d'une manoeuvre pour exécuter cette dernière
1 - Créditer votre compte
2 - Débiter votre compte
3 - Afficher la solde de votre compte
4 - Afficher les 10 dernières opérations effectuées
5 - Quitter la banque

5
paulbouchard@air-de-paul TCP %
```

## Server

```
=====
Affichage des 10 dernières opérations

*****
* Déconnexion de la banque *
*****
paulbouchard@air-de-paul TCP %
```

Le code de la partie UDP est quasiment identique et fonctionne de la même manière donc pas besoin d'en faire des captures d'écran.