

Compte Rendu Projet Réseau

Sommaire :

Introduction	2
Présentation de la BarkouDouzetBanque	2
Architecture du projet	3
Description fonction communication.c	3
Fonction identification	4
Fonction cpt	4
Fonction Compte	4
Fonction DixOperations	5
Fonction itoa	5
Protocole TCP	5
Protocole UDP	7
Algorithme de la communication	7
Jeu de scénarios	8
Scénario 1 :	8
Avec le protocole TCP :	9
Avec le protocole UDP :	13
Scénario2 :	17
Avec le protocole TCP :	17
Avec le protocole UDP :	20

*Lien github contenant tous les codes :
https://github.com/Julian-barkoudeh/tcp_projet

1. Introduction

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

Nous avons utilisé deux protocoles que nous avons vu en cours pour la même application : Le protocole TCP et UDP.

2. Présentation de la BarkouDouzetBanque

Bienvenue à la BarkouDouzetBanque !

Nous allons vous expliquer son fonctionnement !

Chaque client a 3 comptes. Ils ont déjà été initialisés par nos soins.

Voici l'état des comptes de la banque :

- client 1 :

identifiant : julian

mot de passe : 1234

état des comptes :

- compte 1 : 100 euros
- compte 2 : 3000 euros
- compte 3 : vide

- client 2 :

identifiant : camille

mot de passe : 1234

état des comptes :

- compte 1 : 500 euros
- compte 2 : vide
- compte 3 : vide

- client 3 :

identifiant : maria

mot de passe : 1234

état des comptes :

- compte 1 : 10000 euros
- compte 2 : vide
- compte 3 : vide

Quelques mots sur les conditions des contrats qui sont établis dans la banque. Il n'est pas possible d'avoir un solde négatif. Si un compte est vide, il contient la valeur "-1", et cela veut dire que ce compte n'a pas été ouvert. Il est possible de changer de

client, et les opérations seront effectuées sur ses comptes. A tout moment, vous pouvez écrire "exit" afin de quitter la connexion avec le serveur et d'arrêter les codes de serveur et de client.

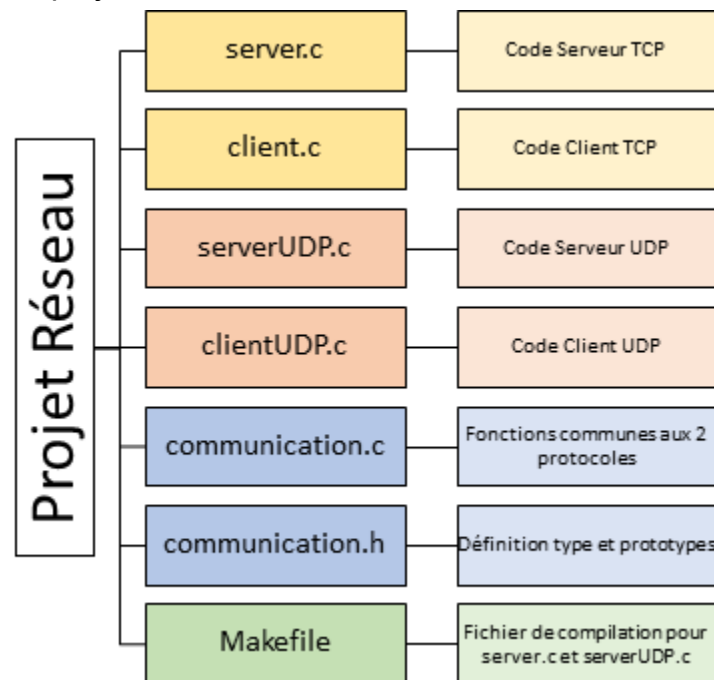
Vous pouvez effectuer 4 types d'opérations :

1. Faire un ajout : Ajouter une somme dans le compte que vous aurez choisi.
2. Faire un retrait : Débitier un montant dans le compte voulu
3. Obtenir le solde de votre compte.
4. Obtenir les 10 dernières opérations du client.

Les chiffres 1 à 4 correspondent aux opérations, donc afin que votre demande soit prise en compte il faut saisir le numéro de la demande. Le serveur vous renverra les réponses à vos requêtes.

3. Architecture du projet

Voici l'architecture du projet :



Les fichiers jaunes correspondent aux codes utiles spécialement pour le protocole TCP.

Les fichiers en orange correspondent au protocole UDP.

Les fichiers bleus sont constitués de fonctions utiles pour les deux prototypes comme la fonction d'identification.

Le makefile en vert facilite la compilation du projet et l'édition de liens des fichiers.

4. Description fonction communication.c

Le script communication.c contient :

- Une fonction pour l'identification: identification
- Une fonction pour récupérer les modifications dans la base de données : cpt
- Une fonction de gestion du compte : compte
- Une fonction pour gérer l'historique d'un client : DixOperations
- Deux fonctions utiles pour la fonction itoa : swap et reverse
- Une fonction de conversion du type int vers str : itoa

Ces fonctions seront utilisées dans le script server.c et serverUDP.c notamment.

Ce sont les fonctions communes aux deux protocoles.

a. Fonction identification

But : Gérer l'identification d'un client avec son mot de passe

Input :

- char* iden = Identifiant du client
- char* mdp = Mot de passe du client
- t_client BDD_id[clients] = Tableau de données des clients de la banque avec leur mot de passe

	Client 1	Client 2	Client 3
Identifiant	julian	camille	maria
Mot de passe	1234	1234	1234

Output : La fonction renvoie :

- 0 : si l'identifiant est mauvais
- 1 : si le mdp est mauvais
- 2 : si l'identification se passe bien

b. Fonction cpt

But : Récupérer les modifications dans la base de données

Input :

- char* iden = Identifiant du client
- t_client BDD_id[clients] = Tableau de données des clients de la banque avec leur mot de passe

Output :

- i = Indice du tableau de base de données où il y a eu des modifications

c. Fonction Compte

But : Fonction d'interaction avec le compte, c'est ici que l'on effectue les modifications sur un compte bancaire.

Input :

- char* iden = Identifiant du client
- int numCpt = Numéro du compte à utiliser

- int BDD_c[3][clients] = Tableau de données des clients et de l'état de leur compte
Il est construit de cette manière :

	Client 1	Client 2	Client 3
Compte 1	x1	x2	x3
Compte 2	y1	y2	y3
Compte 3	z1	z2	z3

- t_client BDD_id[clients] = Tableau de données des clients de la banque avec leur mot de passe.

- int op = Type d'opération courante

- int somme = Montant de l'opération en cas d'ajout ou de retrait

Output :

- BDD_c[numCpt][cpt] = Tableau de base de données mis à jour en fonction des comptes

d. Fonction DixOperations

But : Gérer l'historique des opérations d'un client, cette fonction joue le rôle de livre des comptes.

Input :

- t_chaine operation = Chaîne récapitulative des actions

- int op = Type d'opération courante

- char* numCpt = Numéro du compte utilisé

- char *somme = Montant de l'opération en cas d'ajout ou de retrait

Output :

- t_chaine operation = Chaîne récapitulative des 10 dernières actions avec le compte associé.

e. Fonction itoa

But : Convertir des int en str

Input :

- int num = Nombre à convertir

- char* buffer = Buffer de conversion vers le type str

- int base : base de conversion (decimal, hexadecimal etc)

Output :

- char* buffer = Buffer de conversion vers le type str

5. Protocole TCP

Dans ce protocole la gestion de la connexion ainsi que la création des sockets se passe dans la fonction main. Au début le socket est créé, il s'agit d'abord d'une variable entier qui va être utilisé dans les fonctions d'envoi et de réception.

```
/* Fonction main
But : Fonction main du projet coté serveur avec TCP
*/
int main(int argc, char *argv[])
{
    int socket_desc, client_sock, c, read_size;
    struct sockaddr_in server, client;
    char client_message[2000];

    //Création du socket TCP
    socket_desc = socket(AF_INET, SOCK_STREAM, 0);
    if (socket_desc == -1)
    {
        printf("Could not create socket");
    }
    puts("Socket created");

    //Préparation de la structure sockaddr_in
    server.sin_family = AF_INET;
    server.sin_addr.s_addr = INADDR_ANY;
    server.sin_port = htons(8888);

    //Etablissement de la liaison
    if (bind(socket_desc, (struct sockaddr *)&server, sizeof(server)) < 0)
    {
        // Message d'erreur
        perror("bind failed. Error");
        return 1;
    }
    puts("bind done");

    //Mode écoute
    listen(socket_desc, 3);

    //En attente d'une connexion entrante
    puts("Waiting for incoming connections...");
    c = sizeof(struct sockaddr_in);
```

Afin de lier le client au serveur avec la fonction bind, la structure de socket est remplie avec les informations importantes pour la connexion comme le port qui est 8888 dans ce cas. Le serveur reste en attente d'une connexion d'un client, ensuite enchaîne les opérations d'échange avec le client quand la connexion a été établie.

Une structure doit être respectée ensuite afin de pouvoir envoyer et de recevoir les messages du clients. Un seul buffer peut être utilisé pour faire ces opérations.

Dans le cas de client, les échanges avec le serveur se passent dans une boucle while qui s'exécute à l'infini, tant que le client n'a pas entré "exit" comme message. Tout d'abord un message demandant au client de saisir son message, et il est obtenu avec la fonction "scanf", et est stocké dans le buffer "message". Ensuite ce message est envoyé avec la fonction "send". Cette fonction prend en paramètre le socket créé pour la connexion avec le serveur, le buffer ainsi que la longueur de message. La fonction retourne un entier supérieur à 0 si le message a été bien envoyé. Le client se met après en mode "listening" en exécutant la fonction "rcv". Cette fonction garde la client en écoute sur la voie de connexion avec le serveur d'un nouveau message. Tant que la

fonction n'a pas retourné un entier supérieur à 0, le reste de script ne s'exécute pas. Le message de serveur est ensuite affiché à l'utilisateur, et le processus se répète.

Pour le serveur, la structure de l'échange a été faite un peu différemment.

Les mêmes fonctions d'envoi et de réception sont utilisées, cependant l'ordre et l'emplacement ne sont pas les mêmes. En effet, dans le code de serveur deux buffers : un pour l'envoi et un pour la réception ont été choisis. Ceci rend plus facile de vider les buffers.

Après que le client s'est connecté, un message de bienvenue lui est envoyé. Ce message est simplement copié dans le buffer d'envoi, qui a été vidé auparavant avec la fonction `bezero`.

Tous les échanges avec le client se passent dans la fonction "func", qui contient le boucle à l'infini. Tout d'abord le buffer de réception est vidé, et le serveur se met en mode d'écoute. Quand le client envoie sa réponse, le serveur traite cette réponse dans une série de conditions, et ensuite copie un message spécifique dans le buffer d'envoi pour chaque situation. Après toutes les boucles "if", le message est envoyé au client avec la fonction "send".

6. Protocole UDP

Avec ce protocole, nous avons utilisé la même structure dans le client et le serveur afin d'effectuer les échanges. La création des sockets ainsi que la connexion d'un client à un serveur sont identiques. Néanmoins, les fonctions utilisées pour l'envoi et la réception sont différentes. En effet, en paramètre de ces fonctions, la structure des sockets est importante comme entrée dans les fonctions. Ces structures, comme mentionné auparavant, contiennent les ports et les adresses.

7. Algorithme de la communication

Voici le déroulé globale de la communication :

- Le client doit envoyer un premier message pour initialiser la communication. Le serveur demande au client d'entrer un mot pour débiter l'échange.
- $k=0$: Le serveur de la banque vous demande d'abord votre identifiant.
 - Si l'identifiant existe vous passer à $k=1$.
 - Sinon le serveur vous envoie un message d'erreur et vous redemande votre identifiant.
- $k=1$: Le serveur vous demande votre mot de passe et fait appelle à la fonction ***identification***.
- $k=2$: Traitement des réponses de l'identification :
 - Si l'identifiant n'existe pas, le serveur affiche un message d'erreur et vous repasser à $k=0$.
 - Si l'identifiant existe mais que le mot de passe est erroné. Le serveur vous envoie un message d'erreur et vous restez à $k=2$.

- Si l'identification se passe bien vous passez à k=3.
- k=3 : On vous demande un numéro de compte à sélectionner entre 0, 1 ou 2.
 - Si le compte n'existe pas, on vous demande d'en choisir un nouveau
 - Si le compte existe, le serveur vous demande quelle opération vous souhaitez effectuer et vous passez à k=4.
- k=4 : Vous avez choisi quel type d'opération vous souhaitez faire.
 - Si op = 3, vous avez demandé le solde de votre compte, le serveur s'exécute. Vous passez à k=6.
 - Si op = 4, vous avez demandé l'affichage des 10 dernières opérations, le serveur s'exécute. Vous passez à k=6.
 - Sinon cela veut dire que op = 1 ou 2. Vous avez choisi un retrait ou un ajout sur votre compte. Le serveur vous demande le montant du virement et vous passez à k=5.
- k=5 : Vous avez choisi un retrait ou un ajout d'un certain montant. Vous passez dans tous les cas à k=6.
 - Si op = 1 : On ajoute ce montant à votre compte.
 - Si op = 2 : On retire ce montant à votre compte. Attention, vous ne pouvez pas être à découvert, donc si le solde est insuffisant le serveur vous redemandera un montant à retirer.
- k=6 : Félicitations ! Vous venez de réaliser votre première opération ! Mais vous voulez sûrement effectuer de nouvelles actions... Pas de soucis, vous allez maintenant devoir choisir la prochaine action :
 1. Changer de client
 2. Changer de compte
 3. Une autre opération
 - Vous voulez changer de client. Vous devez confirmer votre choix en envoyant o ou n. Puis vous repassez à k=0.
 - Vous voulez changer de compte. Vous devez entrer le numéro de ce dernier. Vous passez à k=3.
 - Vous voulez faire une nouvelle opération. Vous passez à k=3. C'est une version allégée de k=3, ainsi pas besoin de redire votre numéro de compte.

Vous pouvez évidemment quitter la communication à tout moment en envoyant le mot "exit" au serveur !

8. Jeu de scénarios

Nous avons établi deux scénarios. L'objectif est de tester notre application et sa robustesse. D'abord nous avons voulu nous mettre dans un cas parfait avec un client exemplaire et puis un scénario catastrophe où le client ne se rappelle plus de rien lors de sa connexion !

Scénario 1 :

Dans ce test nous allons démontrer un scénario parfait :

- Le client s'identifie avec un compte,
- Il ajoute et retire de l'argent de ce compte.
- Il affiche ensuite le solde de son compte
- Avant dernière étape : il change de compte pour effectuer une nouvelle opération.
- Finalement, il demande à afficher toutes les opérations puis quitte la connexion.

Avec le protocole TCP :

Terminal du client :

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
barkoudeh@barkoudeh-IdeaPad-L340-15IRH-Gaming:~/Projet_Reseaux/Server/tcp_projet$ ./client
Socket created
Connected

Server reply :
Bienvenue dans la BarkouDouzetBanque ! Appuyez sur n'importe quel bouton pour vous identifier

Enter message : ok
Server reply :
Veuillez saisir votre numero identifiant
Enter message : julian
Server reply :
Veuillez saisir votre mot de passe
Enter message : 1234
Server reply :
Selectionnez votre numéro de compte
Enter message : 0
Server reply :
Operations : Ecrivez
1. Pour ajouter une somme
2. Pour retirer une somme
3. Pour afficher votre solde
4. Pour afficher les dernières 10 op
OP :
Enter message : 1
Server reply :
Montant en euro :
Enter message : 100
Server reply :
Montant ajouté !
Selectionnez votre prochaine operation :
1.Changer de client
2. Changer de compte
3. Une autre opération

Enter message : 3
Server reply :
Confirmez vous cette opertaion ? o|n
Enter message : o
Server reply :
Operations : Ecrivez
1. Pour ajouter une somme
2. Pour retirer une somme
3. Pour afficher votre solde
4. Pour afficher les dernières 10 op
OP :
Enter message : 2
Server reply :
Montant en euro :
Enter message : 50
Server reply :
Montant retiré !
Selectionnez votre prochaine operation :
1.Changer de client
2. Changer de compte
3. Une autre opération

Enter message : 3
```

```

Enter message : 3
Server reply :
Confirmez vous cette opertaion ? o|n
Enter message : o
Server reply :
Operations : Ecrivez
  1. Pour ajouter une somme
  2. Pour retirer une somme
  3. Pour afficher votre solde
  4. Pour afficher les dernières 10 op
  OP :
Enter message : 3
Server reply :
Votre solde est de 150 euro(s)
  Sélectionnez votre prochaine operation :
    1.Changer de client
    2. Changer de compte
    3. Une autre opération

Enter message : 2
Server reply :
Veuillez saisir le numéro de votre compte

Enter message : 1
Server reply :
Operations : Ecrivez
  1. Pour ajouter une somme
  2. Pour retirer une somme
  3. Pour afficher votre solde
  4. Pour afficher les dernières 10 op
  OP :
Enter message : 1
Server reply :
Montant en euro :
Enter message : 500
Server reply :
Montant ajouté !
  Sélectionnez votre prochaine operation :
    1.Changer de client
    2. Changer de compte
    3. Une autre opération

Enter message : 3
Server reply :
Confirmez vous cette opertaion ? o|n
Enter message : o
Server reply :
Operations : Ecrivez
  1. Pour ajouter une somme
  2. Pour retirer une somme
  3. Pour afficher votre solde
  4. Pour afficher les dernières 10 op
  OP :
Enter message : 4
Server reply :
Dernières dix opérations sur vos comptes :
Compte 0: +100

```

```
Compte 0: -50
Compte 0: 150
Compte 1: +500

Sélectionnez prochaine operation :
1.Changer de client
2. Changer de compte
3. Une autre opération

Enter message : exit
barkoudeh@barkoudeh-IdeaPad-L340-15IRH-Gaming:~/Projet_Reseaux/Server/tcp_proje
```

nine

Terminal server :

```
Terminal Help
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
barkoudeh@barkoudeh-IdeaPad-L340-15IRH-Gaming:~/Projet_Reseaux/Server/tcp_projet$ ./tcp.out
Socket created
bind done
Waiting for incoming connections...
Connection accepted
Debut de boucle
From Client : (ok)
Boucle k = 0
To Client: Veuillez saisir votre numero identifiant
Debut de boucle
From Client : (julian)
boucle k = 1
To Client: Veuillez saisir votre mot de passe
Debut de boucle
From Client : (1234)
boucle k = 2
L'identification s'est déroulé avec succès
L'identification s'est déroulé avec succès
L'identification s'est déroulé avec succès
To Client: Sélectionnez votre numéro de compte
Debut de boucle
From Client : (0)
To Client: Operations : Ecrivez
  1. Pour ajouter une somme
  2. Pour retirer une somme
  3. Pour afficher votre solde
  4. Pour afficher les dernières 10 op
OP :
Debut de boucle
From Client : (1)
To Client: Montant en euro :
Debut de boucle
From Client : (100)
boucle k 5
To Client: Montant ajouté !
Sélectionnez votre prochaine operation :
  1.Changer de client
  2. Changer de compte
  3. Une autre opération
Debut de boucle
From Client : (3)
To Client: Confirmez vous cette opertaion ? o|n
Debut de boucle
From Client : (o)
To Client: Operations : Ecrivez
  1. Pour ajouter une somme
  2. Pour retirer une somme
  3. Pour afficher votre solde
  4. Pour afficher les dernières 10 op
OP :
Debut de boucle
From Client : (2)
To Client: Montant en euro :
Debut de boucle
From Client : (50)
boucle k 5
To Client: Montant retiré !
```

```

boucle k 5
To Client: Montant retiré !
Sélectionnez votre prochaine operation :
1.Changer de client
2. Changer de compte
3. Une autre opération

Debut de boucle
From CLient : (3)
To Client: Confirmez vous cette opertaion ? o|n
Debut de boucle
From CLient : (o)
To Client: Operations : Ecrivez
1. Pour ajouter une somme
2. Pour retirer une somme
3. Pour afficher votre solde
4. Pour afficher les dernières 10 op
OP :
Debut de boucle
From CLient : (3)
To Client: Votre solde est de 150 euro(s)
Sélectionnez votre prochaine operation :
1.Changer de client
2. Changer de compte
3. Une autre opération

Debut de boucle
From CLient : (2)
To Client: Veuillez saisir le numéro de votre compte

Debut de boucle
From CLient : (1)
To Client: Operations : Ecrivez
1. Pour ajouter une somme
2. Pour retirer une somme
3. Pour afficher votre solde
4. Pour afficher les dernières 10 op
OP :
Debut de boucle
From CLient : (1)
To Client: Montant en euro :
Debut de boucle
From CLient : (500)
boucle k 5
To Client: Montant ajouté !
Sélectionnez votre prochaine operation :
1.Changer de client
2. Changer de compte
3. Une autre opération

Debut de boucle
From CLient : (3)
To Client: Confirmez vous cette opertaion ? o|n
Debut de boucle
From CLient : (o)
To Client: Operations : Ecrivez
1. Pour ajouter une somme
2. Pour retirer une somme
3. Pour afficher votre solde
4. Pour afficher les dernières 10 op
OP :

```

```

Debut de boucle
From Client : (3)
To Client: Confirmez vous cette opertaion ? o|n
Debut de boucle
From Client : (o)
To Client: Operations : Ecrivez
1. Pour ajouter une somme
2. Pour retirer une somme
3. Pour afficher votre solde
4. Pour afficher les dernières 10 op
OP :
Debut de boucle
From Client : (4)
To Client: Dernières dix opérations sur vos comptes :
Compte 0: +100
Compte 0: -50
Compte 0: 150
Compte 1: +500

Sélectionnez prochaine operation :
1.Changer de client
2. Changer de compte
3. Une autre opération

Debut de boucle
From Client : ()
To Client:
Debut de boucle
From Client : ()
barkoudeh@barkoudeh-IdeaPad-L340-15IRH-Gaming:~/Projet_Reseaux/Server/tcp_projet$ █

```

Avec le protocole UDP :

Terminal client :

```
18 avril 19:47 ●
clientUDP.c - Untitled (Workspace) - Visual Studio Code
in Terminal Help
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
barkoudeh@barkoudeh-IdeaPad-L340-15IRH-Gaming:~/Projet_Reseaux/Server/tcp_projet$ ./clientudp
Bienvenue dans la BarkouDouzetBanque! Appuyez sur n'importe quel bouton pour vous identifier
Enter message : ok
Veuillez saisir votre numero identifiant
Enter message : julian
Veuillez saisir votre mot de passe
Enter message : 1234
Selectionnez votre numéro de compte
Enter message : 0
Operations : Ecrivez
  1. Pour ajouter une somme
  2. Pour retirer une somme
  3. Pour afficher votre solde
  4. Pour afficher les dernières 10 op
OP :
Enter message : 1
Montant en euro :
Enter message : 100
Montant ajouté !
  Sélectionnez votre prochaine operation :
  1.Changer de client
  2. Changer de compte
  3. Une autre opération

Enter message : 2
Veuillez saisir le numéro de votre compte

Enter message : 1
Operations : Ecrivez
  1. Pour ajouter une somme
  2. Pour retirer une somme
  3. Pour afficher votre solde
  4. Pour afficher les dernières 10 op
OP :
Enter message : 2
Montant en euro :
Enter message : 200
Montant retiré !
  Sélectionnez votre prochaine operation :
  1.Changer de client
  2. Changer de compte
  3. Une autre opération

Enter message : 3
Confirmez vous cette opertaion ? o|n
Enter message : o
Operations : Ecrivez
  1. Pour ajouter une somme
  2. Pour retirer une somme
  3. Pour afficher votre solde
  4. Pour afficher les dernières 10 op
OP :
Enter message : 3
Votre solde est de 2800 euro(s)
  Sélectionnez votre prochaine operation :
  1.Changer de client
  2. Changer de compte
  3. Une autre opération
```



```
Enter message : 3
Votre solde est de 2800 euro(s)
Sélectionnez votre prochaine operation :
1.Changer de client
2. Changer de compte
3. Une autre opération
```

```
Enter message : 3
Confirmez vous cette opertaion ? o|n
Enter message : o
Operations : Ecrivez
1. Pour ajouter une somme
2. Pour retirer une somme
3. Pour afficher votre solde
4. Pour afficher les dernières 10 op
OP :
```

```
Enter message : 4
Dernières dix opérations sur vos comptes :
0000|Compte 0: +100
Compte 1: -200
00000Compte 1: 2800
00000900|
```

```
Sélectionnez prochaine operation :
1.Changer de client
2. Changer de compte
3. Une autre opération
```

```
Enter message : exit
```

```
barkoudeh@barkoudeh-IdeaPad-L340-15IRH-Gaming:~/Projet_Reseaux/Server/tcp_projet$ █
```

Terminal server :

```

jn Terminal Help
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

barkoudeh@barkoudeh-IdeaPad-L340-15IRH-Gaming:~/Projet_Reseaux/Server/tcp_projet$ ./clientudp
Bienvenue dans la BarkouDouzetBanque! Appuyez sur n'importe quel bouton pour vous identifier
Enter message : ok
Veuillez saisir votre numero identifiant
Enter message : julian
Veuillez saisir votre mot de passe
Enter message : 1234
Selectionnez votre numéro de compte
Enter message : 0
Operations : Ecrivez
  1. Pour ajouter une somme
  2. Pour retirer une somme
  3. Pour afficher votre solde
  4. Pour afficher les dernières 10 op
OP :
Enter message : 1
Montant en euro :
Enter message : 100
Montant ajouté !
  Sélectionnez votre prochaine operation :
  1.Changer de client
  2. Changer de compte
  3. Une autre opération

Enter message : 2
Veuillez saisir le numéro de votre compte

Enter message : 1
Operations : Ecrivez
  1. Pour ajouter une somme
  2. Pour retirer une somme
  3. Pour afficher votre solde
  4. Pour afficher les dernières 10 op
OP :
Enter message : 2
Montant en euro :
Enter message : 200
Montant retiré !
  Sélectionnez votre prochaine operation :
  1.Changer de client
  2. Changer de compte
  3. Une autre opération

Enter message : 3
Confirmez vous cette opertaion ? o|n
Enter message : o
Operations : Ecrivez
  1. Pour ajouter une somme
  2. Pour retirer une somme
  3. Pour afficher votre solde
  4. Pour afficher les dernières 10 op
OP :
Enter message : 3
Votre solde est de 2800 euro(s)
  Sélectionnez votre prochaine operation :
  1.Changer de client
  2. Changer de compte
  3. Une autre opération
```

```

Enter message : 3
Confirmez vous cette opertaion ? o|n
Enter message : o
Operations : Ecrivez
  1. Pour ajouter une somme
  2. Pour retirer une somme
  3. Pour afficher votre solde
  4. Pour afficher les dernières 10 op
OP :
Enter message : 3
Votre solde est de 2800 euro(s)
Sélectionnez votre prochaine operation :
  1.Changer de client
  2. Changer de compte
  3. Une autre opération

Enter message : 3
Confirmez vous cette opertaion ? o|n
Enter message : o
Operations : Ecrivez
  1. Pour ajouter une somme
  2. Pour retirer une somme
  3. Pour afficher votre solde
  4. Pour afficher les dernières 10 op
OP :
Enter message : 4
Dernières dix opérations sur vos comptes :
0000|Compte 0: +100
Compte 1: -200
00000Compte 1: 2800
00000900|
Sélectionnez prochaine operation :
  1.Changer de client
  2. Changer de compte
  3. Une autre opération

Enter message : exit
barkoudeh@barkoudeh-IdeaPad-L340-15IRH-Gaming:~/Projet_Reseaux/Server/tcp_projet$ █

```

Scénario2 :

Nous allons tester dans ce scénario le cas où le client entre un mauvais identifiant ainsi qu'un mot de passe et un compte qu'il ne possède pas. Nous allons aussi tester le cas où le client demande de retirer un montant plus grand que son solde.

Avec le protocole TCP :

Terminal client :

```
serverUDP.c - Untitled (Workspace) - Visual  
n Terminal Help  
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL  
barkoudeh@barkoudeh-IdeaPad-L340-15IRH-Gaming:~/Projet_Reseaux/Server/tcp_projet$ ./client  
Socket created  
Connected  
  
Server reply :  
Bienvenue dans la BarkouDouzetBanque ! Appuyez sur n'importe quel bouton pour vous identifier  
  
Enter message : ok  
Server reply :  
Veuillez saisir votre numero identifiant  
Enter message : Camille  
Server reply :  
Veuillez saisir votre mot de passe  
Enter message : 1234  
Server reply :  
Identifiant inexistant  
Resaisissez votre identifiant  
  
Enter message : camille  
Server reply :  
Veuillez saisir votre mot de passe  
Enter message : 1111  
Server reply :  
Mauvais mot de passe  
Resaisissez votre mot de passe  
  
Enter message : 1234  
Server reply :  
Selectionnez votre numéro de compte  
Enter message : 1  
Server reply :  
Ce compte n'existe pas, selectionnez un autre compte  
Compte:  
Enter message : 0  
Server reply :  
Operations : Ecrivez  
1. Pour ajouter une somme  
2. Pour retirer une somme  
3. Pour afficher votre solde  
4. Pour afficher les dernières 10 op  
OP :  
Enter message : 2  
Server reply :  
Montant en euro :  
Enter message : 5000  
Server reply :  
Solde insuffisant de 500  
Entrez un montant plus petit :  
Enter message : 100  
Server reply :  
Montant retiré !  
Sélectionnez votre prochaine operation :  
1.Changer de client  
2. Changer de compte  
3. Une autre opération  
  
Enter message : 3  
Server reply :
```

Terminal server :

in Terminal Help

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

barkoudeh@barkoudeh-IdeaPad-L340-15IRH-Gaming:~/Projet_Reseaux/Server/tcp_projet$ ./tcp.out
Socket created
bind done
Waiting for incoming connections...
Connection accepted
Debut de boucle
From CLient : (ok)
Boucle k = 0
To Client: Veuillez saisir votre numero identifiant
Debut de boucle
From CLient : (Camille)
boucle k = 1
To Client: Veuillez saisir votre mot de passe
Debut de boucle
From CLient : (1234)
boucle k = 2
identification identifiant non reussie avec iden :(Camille)
To Client: Identifiant inexistant
Resaisissez votre identifiant

Debut de boucle
From CLient : (camille)
boucle k = 1
To Client: Veuillez saisir votre mot de passe
Debut de boucle
From CLient : (1111)
boucle k = 2
L'identification avec ce mot de passe a échoué
L'identification avec ce mot de passe a échoué
To Client: Mauvais mot de passe
Resaisissez votre mot de passe

Debut de boucle
From CLient : (1234)
boucle k = 2
L'identification s'est déroulé avec succès
L'identification s'est déroulé avec succès
L'identification s'est déroulé avec succès
To Client: Selectionnez votre numéro de compte
Debut de boucle
From CLient : (1)
To Client: Ce compte n'existe pas, selectionnez un autre compte
Compte:
Debut de boucle
From CLient : (0)
To Client: Operations : Ecrivez
  1. Pour ajouter une somme
  2. Pour retirer une somme
  3. Pour afficher votre solde
  4. Pour afficher les dernières 10 op
  OP :
Debut de boucle
From CLient : (2)
To Client: Montant en euro :
Debut de boucle
From CLient : (5000)
boucle k 5
To Client: Solde insuffisant de 500
```

Avec le protocole UDP :

Terminal client :

```
serverUDP.c - Untitled (Workspace) - Visual Studio Code
Terminal Help
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
barkoudeh@barkoudeh-IdeaPad-L340-15IRH-Gaming:~/Projet_Réseaux/Server/tcp_projet$ ./clientudp
Bienvenue dans la BarkouDouzetBanque! Appuyez sur n'importe quel bouton pour vous identifier
Enter message : ok
Veuillez saisir votre numero identifiant
Enter message : mariaa
Veuillez saisir votre mot de passe
Enter message : 1234
Identifiant inexistant
Resaisissez votre identifiant

Enter message : maria
Veuillez saisir votre mot de passe
Enter message : 1111
Mauvais mot de passe
Resaisissez votre mot de passe

Enter message : 1234
Selectionnez votre numéro de compte
Enter message : 2
Ce compte n'existe pas, selectionnez un autre compte
Compte:
Enter message : 0
Operations : Ecrivez
1. Pour ajouter une somme
2. Pour retirer une somme
3. Pour afficher votre solde
4. Pour afficher les dernières 10 op
OP :
Enter message : 2
Montant en euro :
Enter message : 20000
Solde insuffisant de 10000
Entrez un montant plus petit :
Enter message : 5000
Montant retiré !
Sélectionnez votre prochaine operation :
1.Changer de client
2. Changer de compte
3. Une autre opération

Enter message : 3
Confirmez vous cette opertaion ? o|n
Enter message : o
Operations : Ecrivez
1. Pour ajouter une somme
2. Pour retirer une somme
3. Pour afficher votre solde
4. Pour afficher les dernières 10 op
OP :
Enter message : 3
Votre solde est de 5000 euro(s)
Sélectionnez votre prochaine operation :
1.Changer de client
2. Changer de compte
3. Une autre opération

Enter message : 1
Confirmez vous cette opertaion ? o|n
```

Terminal server :

```
serverUDP.c - Untitled (Workspace) - Visual Studio Code
un Terminal Help
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
barkoudeh@barkoudeh-IdeaPad-L340-15IRH-Gaming:~/Projet_Reseaux/Server/tcp_projet$ ./udp.out
Debut de boucle
From Client : (ok)
boucle k = 0
Buffer = Veuillez saisir votre numero identifiant
To Client: Veuillez saisir votre numero identifiant
Debut de boucle
From Client : (mariaa)
boucle k = 1
To Client: Veuillez saisir votre mot de passe
Debut de boucle
From Client : (1234)
boucle k = 2
identification identifiant non reussie avec iden :(mariaa)
To Client: Identifiant inexistant
Resaisissez votre identifiant

Debut de boucle
From Client : (maria)
boucle k = 1
To Client: Veuillez saisir votre mot de passe
Debut de boucle
From Client : (1111)
boucle k = 2
L'identification avec ce mot de passe a échoué
L'identification avec ce mot de passe a échoué
To Client: Mauvais mot de passe
Resaisissez votre mot de passe

Debut de boucle
From Client : (1234)
boucle k = 2
L'identification s'est déroulé avec succès
L'identification s'est déroulé avec succès
L'identification s'est déroulé avec succès
To Client: Selectionnez votre numéro de compte
Debut de boucle
From Client : (2)
To Client: Ce compte n'existe pas, selectionnez un autre compte
Compte:
Debut de boucle
From Client : (0)
To Client: Operations : Ecrivez
1. Pour ajouter une somme
2. Pour retirer une somme
3. Pour afficher votre solde
4. Pour afficher les dernières 10 op
OP :
Debut de boucle
From Client : (2)
To Client: Montant en euro :
Debut de boucle
From Client : (20000)
boucle k 5
problème 1 ici
problème là
To Client: Solde insuffisant de 10000
Entrez un montant plus petit :
```