



Rapport de Projet

Programmation Réseau :

Les Sockets

: Khelifi Bilel

RT2-1



Sommaire

I – Présentation du principe des Sockets

- Introduction aux Sockets
- Principe du Modèle OSI

II- Présentation du Projet

1-But du Projet

2-Langage Utilisé

3-Description Détaillée de L'Application

- Etude des Différents cas .
- Présentation de la Méthode de Correction d'Erreurs : La Parité Matricielle LRC .

4-Contenu de La Base de Donnée

5-Limite de L'Application

I- Présentation du principe des Sockets :

Qu'est-ce qu'une socket?

Il s'agit d'un point de communication bidirectionnel par lequel un processus pourra émettre ou recevoir des informations, alors L'objectif des sockets est de permettre à plusieurs processus, éventuellement distants, de communiquer, c'est à dire d'échanger des informations.

Une socket ressemble beaucoup à un téléphone : c'est l'extrémité d'un canal de communication bidirectionnel. En connectant deux sockets ensemble, on peut faire passer des données entre processus, même entre processus s'exécutant sur des machines différentes, exactement de la même façon qu'on parle à travers le téléphone, une fois qu'on s'est connecté chez quelqu'un d'autre en l'appelant.

La notion de sockets a été introduite dans les distributions de Berkeley (un fameux système de type UNIX, dont beaucoup de distributions actuelles utilisent des morceaux de code), c'est la raison pour laquelle on parle parfois de sockets BSD (*Berkeley Software Distribution*).

Il s'agit d'un modèle permettant la communication inter processus afin de permettre à divers processus de communiquer aussi bien sur une même machine qu'à travers un réseau [TCP/IP](#).

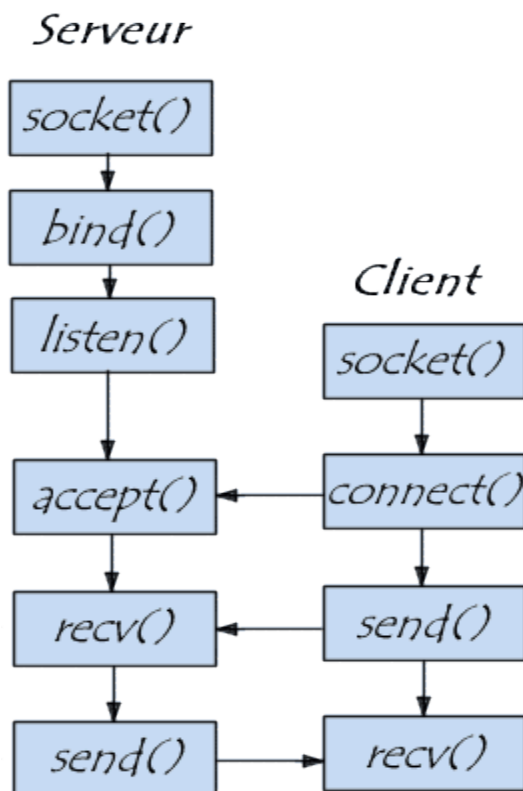
On distingue ainsi deux modes de communication :

- [Le mode connecté](#) (comparable à une communication téléphonique), utilisant le [protocole TCP](#). Dans ce mode de communication, une connexion durable est établie entre les deux processus, de telle façon que l'adresse de destination n'est pas nécessaire à chaque envoi de données.
- [Le mode non connecté](#) (analogue à une communication par courrier), utilisant le [protocole UDP](#). Ce mode nécessite l'adresse de destination à chaque envoi, et aucun accusé de réception n'est donné.

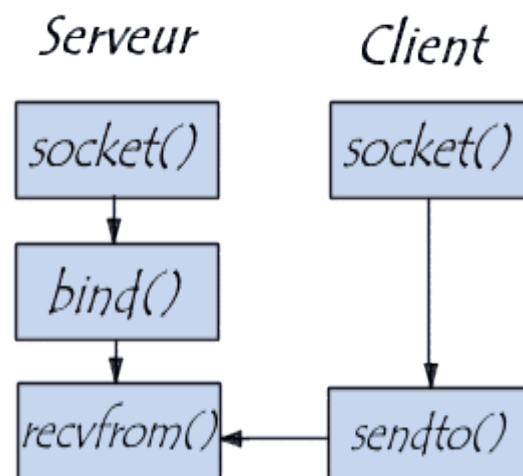
Un serveur doit être à l'écoute de messages éventuels. Toutefois, l'écoute se fait différemment selon que le socket est en mode connecté (TCP) ou non (UDP).

- **En mode connecté**, le message est reçu d'un seul bloc.
Ainsi en mode connecté, la fonction *listen()* permet de placer le socket en mode passif (à l'écoute des messages). En cas de message entrant, la connexion peut être acceptée grâce à la fonction *accept()*. Lorsque la connexion a été acceptée, le serveur reçoit les données grâce à la fonction *recv()*.
- **En mode non connecté**, comme dans le cas du courrier, le destinataire reçoit le message petit à petit (la taille du message est indéterminée) et de façon désordonnée.
Le serveur reçoit les données grâce à la fonction *recvfrom()*.
La fin de la connexion se fait grâce à la fonction *close()*.

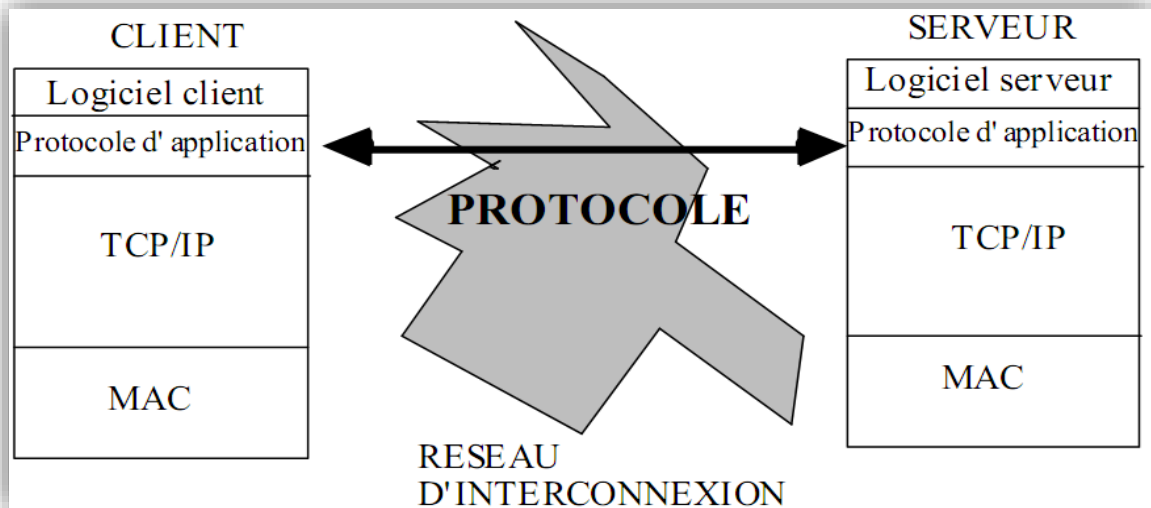
Mode Connecté :



Mode non Connecté :



Principe du Modèle OSI :



I- Présentation du Projet :

1- But

Ce projet est réalisé pour le Développement d'une application Serveur-Client (en java) pour permettre une communication simple : Etudiant-Administration à travers les Sockets.

L'Administration (Serveur), ayant accès aux Données personnelles de chaque Etudiant, essaie de les fournir à l'Etudiant Correspondant via une interface graphique.

2- Langage utilisé :

Cette Application est bien créée avec le langage de programmation Java en ce qui concerne : Client -Serveur et Interface graphique (package java.net) .

D'autre part , pour la gestion de la base de Données , on a utilisé SQLite qui connecte avec le Serveur via Java.jdbc : réalise la Connection Java et sql .

SQLite est une bibliothèque écrite en langage C qui propose un moteur de base de données relationnelle accessible par le langage SQL.

Contrairement aux serveurs de bases de données traditionnels, comme **MySQL** , sa particularité est de ne pas reproduire le schéma habituel client-serveur mais d'être directement intégrée aux programmes. L'intégralité de la base (tables, index ...) est stockée dans un fichier indépendant de la plateforme.

3- Description Détaillée de L' Application :

Principe Général :

Serveur

1. Création du socket serveur en lui attribuant un port.
2. Mise en écoute : attente d'une requête par un client.
3. Accepter la requête en établissant la connexion.
4. Dialoguer avec le client.
5. Fermer la Connexion .

Client :

1. Création du socket client.
2. Demande de requête associée à un port et à une adresse Ip du serveur.
3. Dialoguer avec le serveur.
5. Fermer la Connexion .

Dans notre cas :

Après l'Etablissement de connexion : 1-Serveur & Base de Données

2-Serveur & Client

L'Etudiant est demandé pour entrer son Numéro de Carte Etudiant .

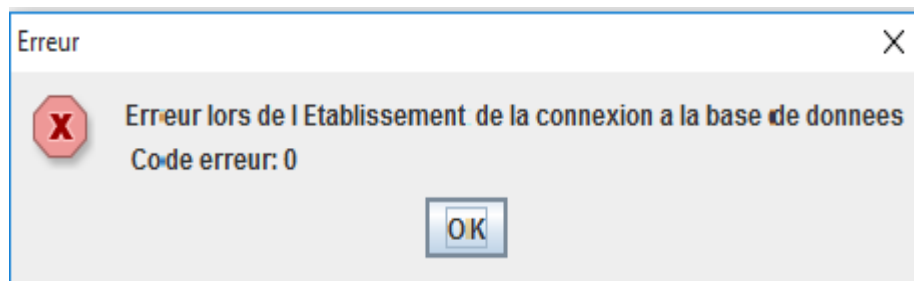
Deux cas se présentent, si ce numéro n'existe pas dans la Base , un message d'Erreur s'affiche et fin du programme , Sinon (Numéro existant) , et par raison d'une simple Vérification , il est demandé une autre fois d'insérer un

numéro_Unique pour éviter que tout Etudiant pourrait accéder simplement aux Données personnelles des autres !

Résultat d'exécution avec Etude de tous les cas possible :

Avant de connecter le Client au serveur ,certains problèmes peuvent nous rencontrer en faisant la liaison avec la base de Données ,parmi eux on peut se trouver dans le cas ou :

-Base de données non créée ou autre problème de Connection

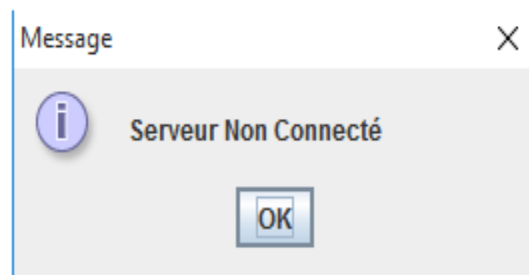


Sinon on affiche :

```
monserveur [Java Application] C:\Program Files (x86)\Java\jre1.8.0_181\bin\javaw.exe
connexion effective
```

On peut maintenant lancer le Client !!

NB : On ne peut pas exécuter le programme client avant que le serveur soit lancé , dans ce cas on affiche :



Après l'exécution des deux programmes, la communication Serveur-Client débute.

Si on essaie d'entrer une Donnée qui n'est pas un entier (180036k) , on répète la demande de Saisie :



Vous etes connecte au serveur

Entrez votre Numero de Carte :

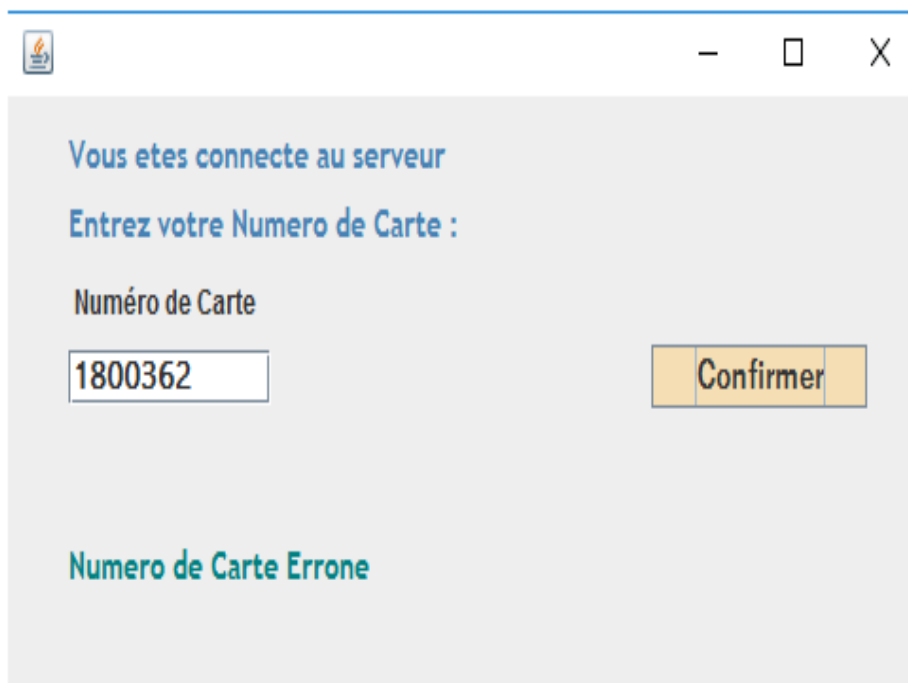
Numéro de Carte

180036k

Confirmer

Réentrez votre Numéro

- ☐ Sinon , et à la saisie d'un numéro qui ne correspond à aucun Etudiant :



Vous etes connecte au serveur

Entrez votre Numero de Carte :

Numéro de Carte

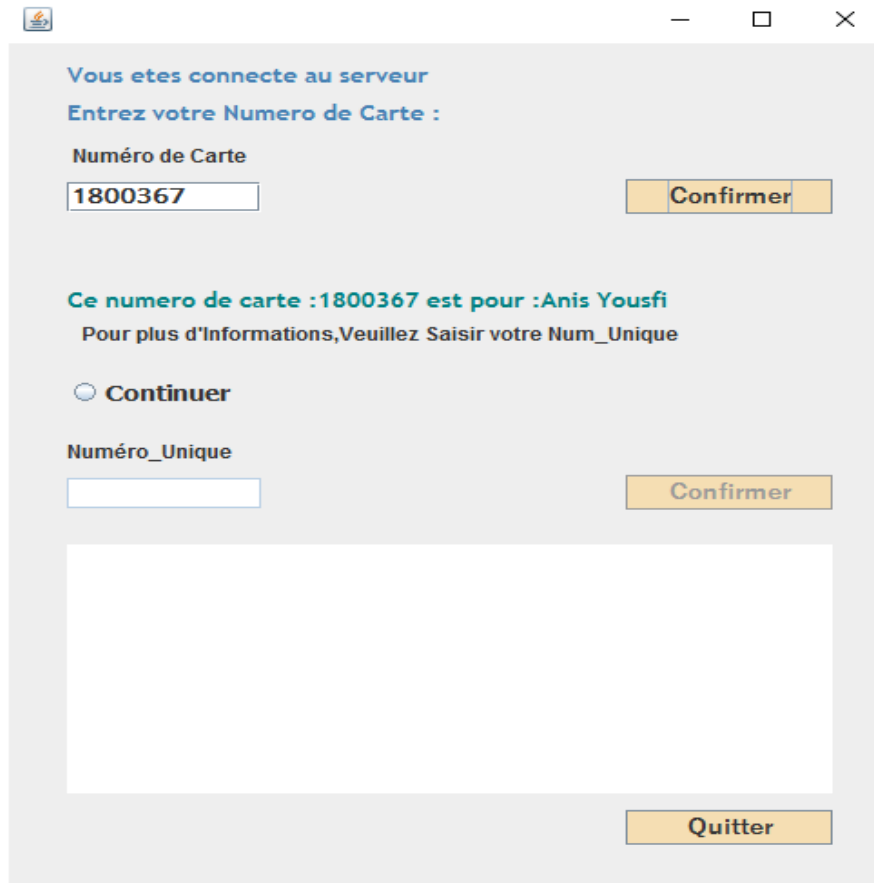
1800362

Confirmer

Numero de Carte Errone

Et fin du pogramme .

- Si notre numéro de carte =1800367 , on affiche l'Etudiant Correspondant et on demande à la fois son numéro unique :



Dans cette Etape ,on ne connait que le Nom et le Prénom de l'Etudiant.S'il ne convient pas à celui qu'on cherche , on peut tout simplement appuyer sur le bouton 'Quitter' , sinon on choisit 'Continuer ' et on saisit son numéro_unique .

En fait , ce numéro sera traité par la méthode de Correction d'Erreur LRC (On va la présenter par la suite) .

En cas où on saisit un numéro erroné (ne correspond pas à cet Etudiant) : On affiche que ce Numéro n'est pas valide et qu'on doit Réessayer .

Ce numero de carte :1800367 est pour :Anis Yousfi
 Pour plus d'Informations,Veuillez Saisir votre Num_Unique

☒ Continuer


Erreur :Reçu =42

Numéro_Unique

Votre Code n'est pas valide
 Vous Devez Reessayer !

Et on a pu, par suite, la protection des données personnelles de cet Etudiant et éviter l'hypothèse qu'elles soient accessibles par d'autres personnes .

- Sinon si tout va bien , on affiche :

 — □ ×

Vous etes connecte au serveur
Entrez votre Numero de Carte :

Numéro de Carte

Ce numero de carte :1800367 est pour :Anis Yousfi
 Pour plus d'Informations,Veuillez Saisir votre Num_Unique

☒ Continuer

Pas d'Erreur lors de la Transmission

Numéro_Unique

Anis Yousfi, 21 ans de Nationalité Tunisienne.
 Ville :Tunis
 Rue :Centre Urbain
 Niveau :Cycle Ingenieur
 Groupe :RT3_1
 Moyenne de l'annee précédente :12
 Moyenne Bac :15
 Email :Yousfi@insat.u-carthage.tn

La Méthode de Correction LRC :

Lorsqu'on envoie le code(numéro_unique) qui varie entre '00' et '99', deux cas sont possibles .Soit le code est envoyé correctement , on affiche alors :



Ce qui indique que les 24 bits envoyés ont été bien reçus .

Dans le cas contraire ,on affiche ceci :



C'est-à-dire , qu'on a :

- ❑ reçu le numéro :15 (malgré qu'on a envoyé '05')
- ❑ détecté l'Erreur par la création d'une Matrice $M(8,3)$.
- ❑ Corrigé cet Erreur par la détection de la colonne et de la ligne correspondante pour avoir finalement le même code envoyé '05' .

NB : Le taux d'Erreur est un peu élevé :environ 80% ,il est généré par une Méthode : **createError** qui choisit un bit bien particulier pour le modifier. (On ne modifie pas les bits de la dernière colonne ou ligne VRC ou LRC donc sur 24 bits au total ,on peut modifier chaque fois 1 seul bit sur 16 bits) .

- ❑ Cette méthode prend comme argument le code envoyé par le client, lance un valeur aléatoire ($e < 1$), s'elle est inférieure à 0.8 ,on choisit une position bien particulière pour la modifier .

```

static public String createError(String a) throws Exception{
    char[] m = a.toCharArray();
    String s;int p=24;
    while((p==7) || (p>=15)){
        double e=Math.random();double pos=e*30;
        if(e<0.8)
            p=(int)pos;
        else return a;
    }
    System.out.println("p= "+p);
    if(p!=-1)
        {if (m[p]=='0') m[p]='1';
         else m[p]='0';}

    s= new String(m);

    return s;}

```

- Passons Maintenant à la Correction qui sera assurée par une méthode : **fixError** (elle n'est pas appelée qu'en cas d'Erreur) ,qui prend en argument une matrice M(8-3) et vérifie le nombre de 1 dans chaque ligne et colonne s'il convient ou pas aux codes LRC et VRC , puis modifie le bit indiqué par ses coordonnées (e2 ,e1) comme c'est clair dans le code marqué en Jaune .

Finalement ,elle retourne une matrice M(8-3) qui sera traitée par d'autres méthodes pour la transformer en 1^{er} lieu en un code de 24 bits et puis passer de ce code qui contient 8 bits de contrôle en un autre de 16 bits et on peut par la suite retrouver notre Numéro_unique envoyé (assurée par la méthode **decoder**)

```
//CORRECTION
public static char[][] fixError(char[][] m){
    int nb=0;int e1=-1;int e2=-1;

    for(int i=0;i<7;i++){
        if(m[i][0]=='1')nb++;}
    nb=nb%2;
    int a = (int) m[7][0]-48;
    if (a!=nb)e1=0;else e1=1;
    nb=0;
    for(int i=0;i<6;i++){
        for(int j=0;j<2;j++){
            if(m[i][j]=='1')nb++;}
        nb=nb%2;
        a = (int) m[i][2]-48;
        if (a!=nb){e2=i;
            i=6;}}

    if( m[e2][e1]=='0')m[e2][e1]='1';
    else m[e2][e1]='0';

    return m;
}
```

4- Contenu de la Base de Donnée :

Table Etudiant :

NUMERO_CARTE	NUMERO_UNIQUE	NUMERO	NOM_PRENOM	AGE	NATIONALITE	EMAIL
1800363	1	1	Laurent Grillote	20	Française	Laurent@insat.u-carthage.tn
1800364	2	2	Christelle Gaulin	21	française	Gaulin@insat.u-carthage.tn
1800365	3	3	Clarlice Blondlot	21	française	Blondlot@insat.u-carthage.tn
1800366	4	4	Karim Khelifi	21	Tunisienne	Khelifi@insat.u-carthage.tn
1800367	5	5	Anis Yousfi	21	Tunisienne	Yousfi@insat.u-carthage.tn
1800368	6	6	Mohammed Mrabet	21	Tunisienne	Mrabet@insat.u-carthage.tn
1800369	7	7	Malak Younsi	21	Tunisienne	Younsi@insat.u-carthage.tn
1800370	8	8	Yassine Melki	21	Tunisienne	Melki@insat.u-carthage.tn

Table Adresse:

NUMERO	VILLE	RUE
1	Lyon	Rue_République
2	Lyon	Rue_République
3	Paris	Banaudon
4	Tunis	Indépendance
5	Tunis	Centre Urbain
6	Tunis	Centre Urbain nord
7	Tunis	Palestine
8	Tunis	Palestine

Table Fiche_scolaire :

NUMERO_CARTE	NIVEAU	FILIERE GROUPE	MOYENNE_ANNEE_PRECEDENTE	MOYENNE_BAC
1800363	Cycle preparatoire	RT2_1	11	14
1800364	Cycle preparatoire	RT2_2	12	11
1800364	Cycle Ingenieur	RT3_1	12	15
1800366	Cycle Ingenieur	RT3_1	12	15
1800367	Cycle Ingenieur	RT3_1	12	15
1800368	Cycle Ingenieur	RT3_1	12	15
1800369	Cycle Ingenieur	RT3_1	12	15
1800370	Cycle Ingenieur	RT4_1	11	12

5-Limite de L' Application:

- Pas de Gestion de différents processus au même Temps : Code Statique , c'est-à-dire qu'on ne peut pas lancer plusieurs fenêtres simultanément (**Pas de Gestion des Threads**) , du coup , en cas où on saisit un numéro différent de celui qu'on veut chercher , on doit seulement Quitter et Relancer notre serveur , puis le Client et Réessayer une autre fois ➔ Donc pour une exécution du code du Serveur , un seul Etudiant peut se Connecter à se dernier .