

Projet Système/Réseaux

Réalisée par:

DIALLO Mariama

BOUKRIS Walid

KABAD Soufiane

SOW Ousmane

1. Introduction.

Dans ce projet, on était censé réaliser une application client/serveur permettant au client d'entrer une requête train et consulter des listes de trains disponibles correspondante à cette requête!

Vu la durée qu'on a (1 semaine) et qu'on est 4 étudiants, l'organisation était notre priorité, on a fait 2 équipes:
Une 1er qui est chargée de la partie Réseau (client/serveur) + Tests sur la partie modèle et gestion des données, et la 2eme qui développe le côté modèle d'application:

- Récupération des données sur les trains d'un fichier texte.
- Stockage de ces données de façon à pouvoir les comparer avec la requête de client.
- Renvoyer le résultat attendu par le client.
- les testes sur la partie Réseau.

2. Utilisation:

● Compilation:

On a fait un makefile qui nous permet de compiler chaque fichier indépendamment.

Exemples: make serveur, make client.

et de supprimer tous les exécutable avec:
make clean.

- **Exécution:**

Pour lancer le serveur: `./serveur <port>`

Pour lancer le client: `./client <host> <port>`

3. Protocole d'échange de client/serveur .

- **Serveur** : Le serveur crée une socket , et attend une demande de connexion des clients, après avoir accepté une connexion d'un client , il crée un processus fils qui va s'occuper de la communication avec le client et il attend d'autres demandes de connexion.

- **Client** : Le client crée une socket , et il envoie une demande de connexion au serveur .
Après que le serveur ait accepté la demande de connexion du client, il envoie une requête composite au serveur et attend de recevoir la réponses puis les affiche à l'écran.

4. Structures et Méthodes utilisées:

- **Structures utilisées:**

On a décidé de créer une structure qui nous permet de stocker toutes les informations à propos d'un trajet de train. Comme numéro, ville départ/arrivée, heure départ/arrivée, prix, promo.

Pour savoir plus on vous invite à voir le fichier trajet.h (très bien commenté).

- **Méthodes utilisées:**

- `int durre(char * strh1, char * strh2)`: Renvoie la durée entre l'heure de départ et l'heure d'arrivée.
- `int comparerHeure(char * strh1, char * strh2)`: Renvoie 1 si l'heure `strh1 < strh2` sinon 0.
- `trajet readtrajet(char *line)`: Prend une chaîne de caractères et renvoie un trajet avec tous les champs bien rempli.
- `char * toString(struct trajet t)`: Prend un trajet et renvoie une chaîne de caractères qui correspond à ce trajet.
- `int comparer(struct trajet t1, struct trajet t2)`: Renvoie 1 si les deux trajets sont les mêmes.

5. Tests.

Durant le projet on a fait plusieurs tests pour vérifier le bon fonctionnement de chaque fonctionnalités, et aussi des tests après l'intégration des différentes tâches:

➤ Test Réseau coté client:

```
ousmane@ousmane-Inspiron-3542: ~/Musique/Projet_System_Reseaux-20191205T201519Z-0...
Fichier Édition Affichage Rechercher Terminal Aide
ousmane@ousmane-Inspiron-3542:~/Musique/Projet_System_Reseaux-20191205T201519Z-0...
01/Projet_System_Reseaux$ ./client localhost 2007
Socket successfully created..
connect successfully created..
Bonjour ! Comment ca va ?

Le test pour la connexion marche très bien, on passe a l'étape suivante !

Bien
```

➤ Test Réseau coté serveur:

```
ousmane@ousmane-Inspiron-3542: ~/Musique/Projet_System_Reseaux-20191205T201519Z-0...
Fichier Édition Affichage Rechercher Terminal Aide
ousmane@ousmane-Inspiron-3542:~/Musique/Projet_System_Reseaux-20191205T201519Z-0...
01/Projet_System_Reseaux$ ./serveur 2007
Socket successfully created..
bind successfully created..
listen successfully created..
accept successfully created..
Bonjour ! Comment ca va ?

Le test pour la connexion marche très bien, on passe a l'étape suivante !

Bien
```

- Test avec Ville de depart , Ville d'arrivée et heure de départ:

```
ousmane@ousmane-Inspiron-3542: ~/Bureau/Projet_System_Reseaux-20191205T201519Z-001/Projet_Syste...
Fichier Édition Affichage Rechercher Terminal Aide
tem_Reseaux$ make client
gcc client.c -o client
ousmane@ousmane-Inspiron-3542:~/Bureau/Projet_System_Reseaux-20191205T201519Z-001/Projet_Sys
tem_Reseaux$ ./client localhost 4000
Socket successfully created..
connect successfully created..
Entrer vos caracteristique de train sous la forme :
Numero;VilleDepart;VileArrivee;HeureDepartMin;HeureDepartMax;Prix;Promo
si vous ne voulez pas remplir un champ mettez X!

*****
X;Paris Gare de Lyon;Valence;10:00;X;X;X;
*****voulez-vous que ça soit trier par rapport au prix(p)/dure(d)*****
p
*****Voici nos resultats*****
9713 Paris Gare de Lyon Valence 10:07 12:19 90.00
6063 Paris Gare de Lyon Valence 10:07 13:15 92.20
*****
Encart en route
```

- Test avec Ville de départ , Ville d'arrivée et tranche horaire de départ:

```
X;Paris Gare de Lyon;Valence;7:00;11:00;X;X
*****voulez-vous que ça soit trier par rapport au prix(p)/dure(d)*****
d
*****Voici nos resultats*****
9713 Paris Gare de Lyon Valence 10:07 12:19 90.00
6063 Paris Gare de Lyon Valence 10:07 13:15 92.20
6035 Paris Gare de Lyon Valence 7:41 10:11 90.40 REDUC
*****
```

➤ **Test avec Ville de départ , Ville d'arrivée :**

```
X;Montelimar;Valence;X;X;X;X
*****voulez-vous que ça soit trier par rapport au prix(p)/dure(d)*****
d
*****Voici nos resultats*****
86175 Montelimar Valence 7:10 7:30 10.78 SUPPL
86174 Montelimar Valence 6:30 6:55 9.80
86177 Montelimar Valence 7:30 7:55 9.80
86178 Montelimar Valence 8:00 8:25 9.80
86179 Montelimar Valence 8:30 8:55 7.84 REDUC
*****
```

➤ **Test avec la promotion:**

```
X;X;X;X;X;X;SUPPL
*****voulez-vous que ça soit trier par rapport au prix(p)/dure(d)*****
p
*****Voici nos resultats*****
86183 Valence Montelimar 14:10 14:30 10.78 SUPPL
86175 Montelimar Valence 7:10 7:30 10.78 SUPPL
17525 Grenoble Valence 16:55 17:55 19.36 SUPPL
*****
```

➤ **Test avec des trajets qui ne sont pas dans le fichier test:**

```
17525;Valence;Geneve;8:00;16:00;X;X  
  
*****voulez-vous que ça soit trier par rapport au prix(p)/dure(d)*****  
  
d  
  
*****Voici nos resultats*****  
  
Pas de resultats  
  
*****
```

6. Conclusion:

Réaliser ce projet nous a apportée des avantages qui sont entre autre:

- Améliorer nos compétences en C, bien maîtriser la gestion des tableaux et l'allocation en mémoire.
- Bien s'organiser en groupe, se partager les tâches, respecter le cahier de charges et respecter les délais.
- bonne utilisation de git, makefile.
- Améliorer nos compétences en Réseau et bien maîtriser les notions de bases.