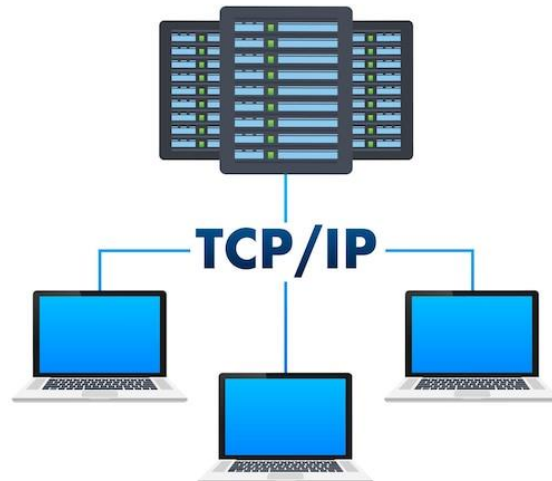


Projet Système/Réseaux
Application client/serveur TCP/IP



Professeur référente : **Mme PIERRE Laurence**

Rapport de projet pour le 02/12/2022

Soutenu le 02/12/2022

L3 MIAGE GROUPE3:

BENABBOU Anas

EL KBABTY Mouad

BOUHALI Walid

MESOUAK Salaheddin

Sommaire :

1) Introduction :	3
2) Réalisation du noyau serveur :	4
3) Réalisation du noyau client :	5
4) Le protocole d'échange entre le client et le serveur :	6
4.1 Le protocole	6
4.2 Résultat :	7
4.2 Fonction pour la gestion de la mort des fils :	8
5) Première requête :	8
5.1 Protocole d'échange de cette requête :	9
5.2 Fonction de recherche :	10
5.3 Résultats :	10
6) Deuxième requête :	11
6.1 Protocole d'échange de cette requête :	11
6.2 Fonction de recherche :	12
6.3 Résultats :	12
7) Troisième requête :	12
7.1 Protocole d'échange de cette requête :	13
7.2 Fonction de la recherche :	13
7.3 Résultats :	14
8) Compilation et exécution:	14
9) Limitations:	15
10) Conclusion:	16

1) Introduction :

Ce rapport présente une description du projet Système/Réseaux d'une application simplifiée client/serveur TCP/IP. Le projet consiste donc à étudier les différents modules destinés à la réalisation d'une application répondant aux besoins forts des utilisateurs en termes de supervision du réseau dans son ensemble. Cette application doit permettre aux utilisateurs d'interroger le serveur sur les différents livres contenu dans sa base de données et de renvoyer les réponses correspondantes au critère fourni. Ce projet nous permettra d'appliquer les notions théoriques vues en réseau et de les appliquer à une utilisation plus pratique.

L'objectif final étant d'établir une connexion durable le temps de la requête entre le client et le serveur. Il devra être capable de fournir au départ le titre, le nom d'auteur, le genre et s'il comporte plus ou moins 300 pages selon la référence transmise par le client, dans le cas où celle-ci n'existe pas le serveur envoie un message informant le client et attend une réponse. A terme le serveur doit pouvoir renvoyer plusieurs livres différents étant donné certaines contraintes (même auteur, nombre de pages, appréciation, mots clés).

Notre premier travail a consisté en une réflexion d'analyse du sujet avant de se lancer dans l'implémentation. Nous avons donc dans un premier temps, commencé par bien étudier les informations fournis dans le memento pour récupérer le maximum d'informations possibles.

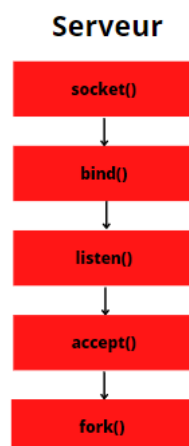
2) Réalisation du noyau serveur :

Du côté du serveur, on crée également une socket en faisant appel à la primitive **socket**, plus précisément **socket (AF_INET, SOCK_STREAM, 0)**, où AF_INET définit le format des adresses possibles et l'ensemble des sockets avec lesquelles notre socket pourra communiquer et dans notre cas c'est pour Internet, SOCK_STREAM pour un échange de séquences continues en mode connecté, et pour le protocole on a donné la valeur 0.

Puis on établit la liaison de la socket d'écoute à l'adresse défini via le **bind, bind (serveur_SOCKET, (struct sockaddr *)&adresse_serveur, taille)** où serveur_SOCKET est le résultat de socket, (struct sockaddr *)&adresse_serveur « un pointeur sur la structure qui correspond à la adresse du serveur » et taille est la taille d'adresse_serveur. On ouvre le service en définissant un nombre limité de connexions en attente (5) via la primitive listen, **listen (serveur_SOCKET, 5)**.

Après on crée l'adresse du client et on stocke sa taille. On entre dans une boucle infinie dans laquelle on accepte une connexion sur la socket d'écoute via la primitive accept, **accept (serveur_SOCKET, (struct sockaddr *)&adresse_client, &taille_adresse)** où serveur_SOCKET est le résultat de socket, (struct sockaddr *)&adresse_serveur « un pointeur sur la structure qui correspond à la adresse du client » et &taille_adresse est une adresse d'un entier qui recevra au retour la taille de l'adresse client.

Au moment où il accepte une demande de connexion le serveur effectue un **fork ()** contenant une boucle infinie permettant la discussion avec le client tant que l'on reçoit des requêtes. Le serveur père quant à lui se met immédiatement en écoute pour renouveler l'opération précédente avec autant de fils nécessaire qu'il y a de clients en attente d'une connexion. Après avoir établie les squelettes du serveur et des clients nous avons testé la communication entre plusieurs clients simultanément avec succès.



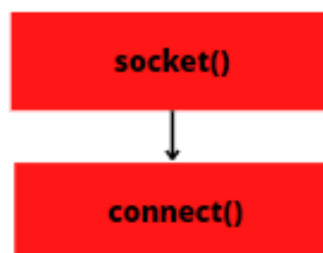
3) Réalisation du noyau client :

Et du coté de client, on a déclaré l'Host et le Port. Après on crée également une socket en faisant appel à la primitive **socket**, plus précisément **socket (AF_INET, SOCK_STREAM, 0)**. Après on a récupérer les informations concernant le serveur avec la primitive **gethostbyname (host)**.

Ensuite, on a affecté les arguments avec la fonction **memcpy**. Cette fonction permet de copier un bloc de mémoire spécifié par le paramètre source, et dont la taille est spécifiée via le paramètre size, dans un nouvel emplacement désigné par le paramètre destination dans notre cas **memcpy (&client_sockaddr.sin_addr (destination), serveur_name->h_addr (source), sizeof(u_long) (size))**.

Après on a connecté le client sur le serveur, et on a pu faire cela à l'aide de la primitive **connect**, plus précisément **connect (client_socket, (struct sockaddr *)&client_sockaddr, sizeof(client_sockaddr))**.

Client

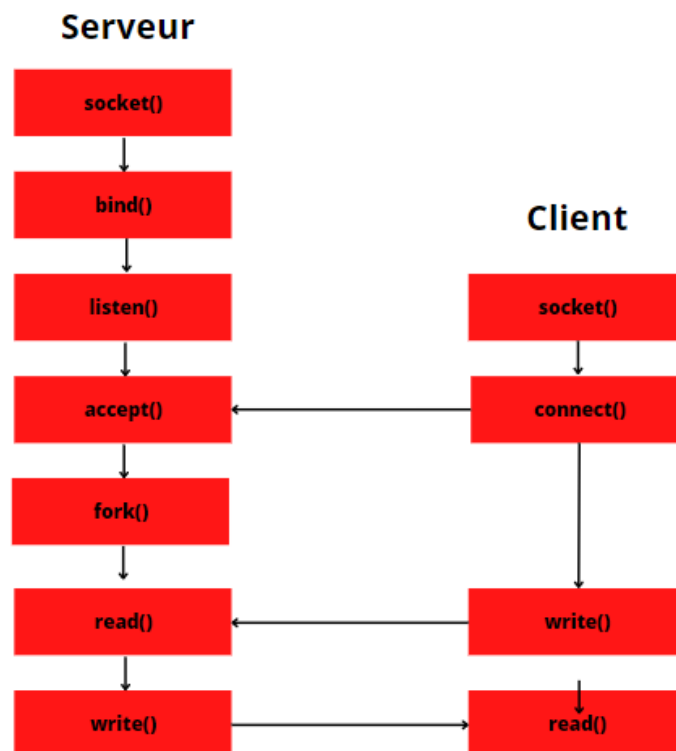


4) Le protocole d'échange entre le client et le serveur :

Pour le dialogue entre le client et le serveur se fait par les fonctions **write** et **read** et aussi à l'aide des sockets

-Pour l'envoi ça se fait avec la fonction **write**.

-Et pour la réception ça se fait avec la fonction **read**.

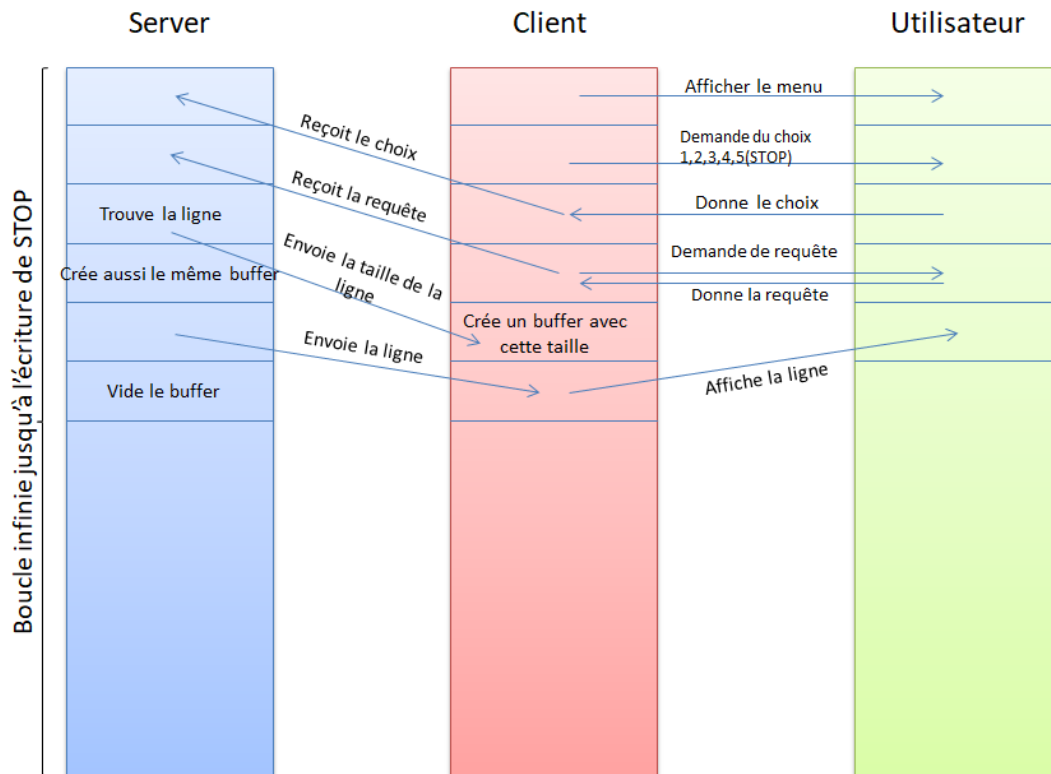


4.1 Le protocole

Au début le client affiche à l'utilisateur le menu, et demande à lui de taper un choix 1, 2, 3, 4 et pour quitter il faut taper 5. Après que l'utilisateur tape son choix et sa requête, le serveur reçoit les deux et trouve la ligne dans le fichier **donné en argument** à l'appel. Le serveur envoie la taille de la ligne au client où on crée un buffer avec cette taille et on crée aussi le même buffer dans le serveur. Ensuite on envoie la ligne de serveur au client où on traite le formatage de données avec des fonctions d'affichage et les envoient à l'utilisateur.

Sur la capture ci-dessus nous pouvons voir qu'après la connexion le client propose 3 choix à l'utilisateur (1 choix par requête) et lui propose d'entrer 4 s'il veut quitter le serveur.

Au cas où l'utilisateur essaye de rentrer un choix inexistant nous l'informons que son choix est non valable et on redemande un nouveau choix.



4.2 Résultat :

On a réussi à connecter plusieurs clients au même serveur, comme afficher ci-dessous :

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
benabban@lm2ag-turing:~/L3MIAGE/SYSRES/groupe3_renduFinal] : ./server 9669 livre.txt
socket d'ouverture ouvert
liason établie avec succès
service ouvert avec succès
A l'écoute ...
Le serveur attend une connexion...
Connexion établie sur le port : 9669, Identificateur du nouveau socket: 4
Processus principale serveur attend une connexion...
Le fils a été créé 444730
connexion établie sur le port : 9669, Identificateur du nouveau socket: 4
Processus principale serveur attend une connexion...
Le fils a été créé 444833
Connexion établie sur le port : 9669, Identificateur du nouveau socket: 4
Processus principale serveur attend une connexion...
Le fils a été créé 444947
Mort du fils
[]

benabban@lm2ag-turing:~/L3MIAGE/SYSRES/groupe3_renduFinal] : ./client lm2ag-turing 9669
TCP server a été bien créée.
Le client est connecté au serveur.
-----MENU-----
Bienvenue dans le serveur
1->livre depuis sa reference
2->livre par mots clés :
3->Savoir la reference et le titre depuis l'auteur et le genre
4-> Tapez 5 pour sortir

Quel requête voulez vous le faire
(Taper 5 pour sortir)
choix requete : 5

Merci pour votre visite , a bientôt
Le client a quitté le serveur .
benabban@lm2ag-turing:~/L3MIAGE/SYSRES/groupe3_renduFinal] : []

benabban@lm2ag-turing:~/L3MIAGE/SYSRES/groupe3_renduFinal] : ./client lm2ag-turing 9669
TCP server a été bien créée.
Le client est connecté au serveur.
-----MENU-----
Bienvenue dans le serveur
1->livre depuis sa reference
2->livre par mots clés :
3->Savoir la reference et le titre depuis l'auteur et le genre
4-> Tapez 5 pour sortir

Quel requête voulez vous le faire
(Taper 5 pour sortir)
choix requete : 956
Choix introuvable , Tapez un nouveau choix ou bien (Taper 5 pour sortir)
choix: 20
Choix introuvable , Tapez un nouveau choix ou bien (Taper 5 pour sortir)
choix: 98
Choix introuvable , Tapez un nouveau choix ou bien (Taper 5 pour sortir)
choix: 3
Quel est auteur : []

```

4.2 Fonction pour la gestion de la mort des fils :

Comme nous pouvons voir sur la capture d'écran après que chaque client quitte le serveur nous avons bien pensé à gérer la mort des fils créés pour chaque client. Nous avons fait le choix de faire cette manipulation en appelant la fonction **sigaction** à qui on a donné un handler (contient la fonction **wait** qui prend un int en paramètre) et bien sûr on lui a donné le flag SA_RESTART pour ne pas bloquer les appels des fonctions primitives appelés après.

```

benabbar@im2ag-turing:~/L3MIAGE/SYSRES/Groupe3_renduFinale$ ./server 9669 livre.txt
socket d'ouverture ouvert
liason établie avec succès
service ouvert avec succès
A l'écoute ...
Le serveur attend une connexion...
Connexion établie sur le port : 9669, Identificateur du nouveau socket: 4
Processus principale serveur attend une connexion...
Le fils a été créé 438958
Connexion établie sur le port : 9669, Identificateur du nouveau socket: 4
Processus principale serveur attend une connexion...
Le fils a été créé 438198
Connexion établie sur le port : 9669, Identificateur du nouveau socket: 4
Processus principale serveur attend une connexion...
Le fils a été créé 438598
Mort du fils
Mort du fils
Mort du fils
benabbar@im2ag-turing:~/L3MIAGE/SYSRES/Groupe3_renduFinale$ ./client im2ag-turing 9669
TCP server a été bien créé.
Le client est connecté au serveur.
-----MENU-----
MENU-----
Bienvenue dans le serveur
1->livre depuis sa reference
2->livre par mots clés :
3->Savoir la reference et le titre depuis l'auteur et le genre
4-> Tapez 5 pour sortir
Quel requête voulez vous le faire
(Taper 5 pour sortir)
choix requete : 5
Merci pour votre visite , a bientôt
Le client a quitté le serveur .
benabbar@im2ag-turing:~/L3MIAGE/SYSRES/Groupe3_renduFinale$
benabbar@im2ag-turing:~/L3MIAGE/SYSRES/Groupe3_renduFinale$ ./client im2ag-turing 9669
TCP server a été bien créé.
Le client est connecté au serveur.
-----MENU-----
MENU-----
Bienvenue dans le serveur
1->livre depuis sa reference
2->livre par mots clés :
3->Savoir la reference et le titre depuis l'auteur et le genre
4-> Tapez 5 pour sortir
Quel requête voulez vous le faire
(Taper 5 pour sortir)
choix requete : 5
Merci pour votre visite , a bientôt
Le client a quitté le serveur .
benabbar@im2ag-turing:~/L3MIAGE/SYSRES/Groupe3_renduFinale$
benabbar@im2ag-turing:~/L3MIAGE/SYSRES/Groupe3_renduFinale$ ./client im2ag-turing 9669
TCP server a été bien créé.
Le client est connecté au serveur.
-----MENU-----
MENU-----
Bienvenue dans le serveur
1->livre depuis sa reference
2->livre par mots clés :
3->Savoir la reference et le titre depuis l'auteur et le genre
4-> Tapez 5 pour sortir
Quel requête voulez vous le faire
(Taper 5 pour sortir)
choix requete : 5
Merci pour votre visite , a bientôt
Le client a quitté le serveur .
benabbar@im2ag-turing:~/L3MIAGE/SYSRES/Groupe3_renduFinale$

```

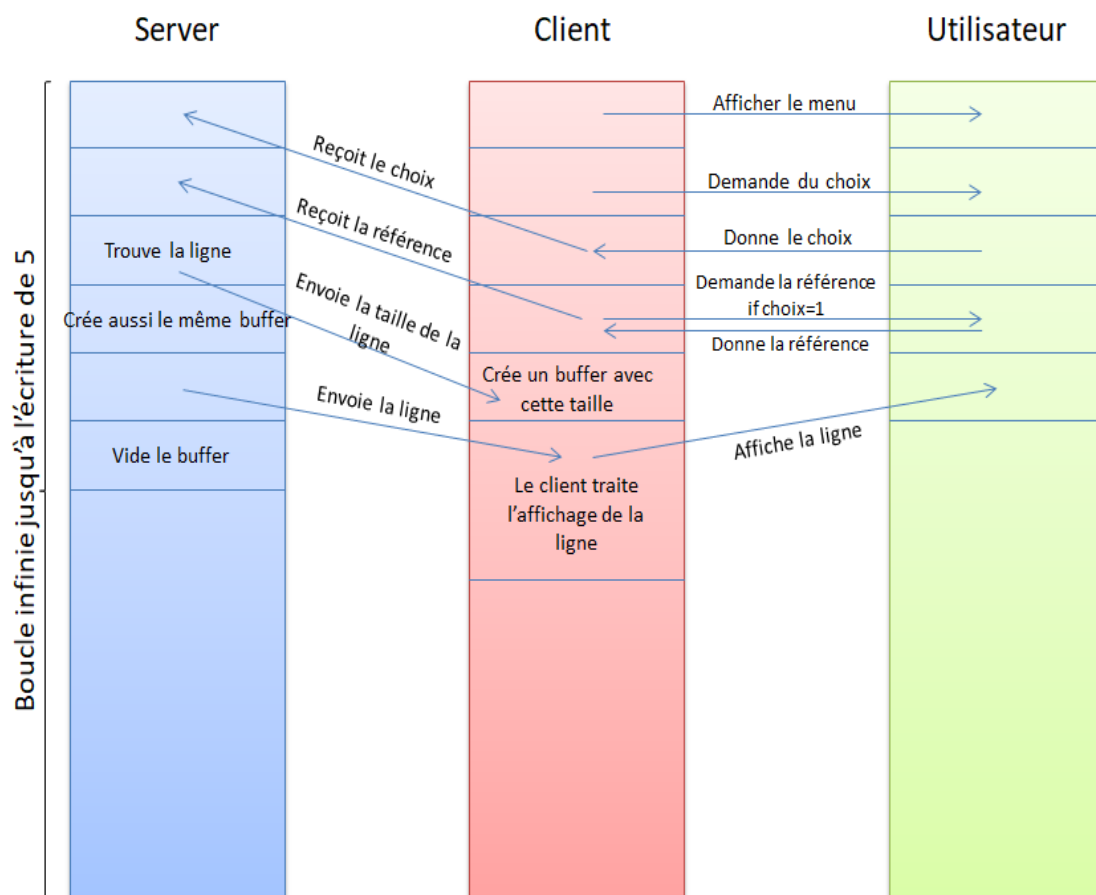
Le protocole d'échange sera expliqué pour chaque cas des 3 cas.

5) Première requête :

Le client soumet la référence d'un livre, et reçoit comme affichage le nom de l'auteur de ce livre, son titre, son genre, et si le nombre de pages est supérieur ou inférieur à 300 pages. Si le client soumet une référence inexistante, on lui affiche un message qui indique que la référence est inexistante et on lui redemande de saisir une référence existante tant qu'il ne tape pas 5.

5.1 Protocole d'échange de cette requête :

Après l'envoi du menu par le client, il attend le retour du choix 1, le client envoie ce choix au serveur ensuite ce dernier se place dans le case de référence. Le client va demander à l'utilisateur la référence souhaitée, ensuite il l'envoie au serveur afin d'appeler la fonction qui va chercher dans le fichier texte la ligne qui contient la référence reçue et stocke la ligne dans une structure Ligne avec la taille de cette ligne. Le choix d'utiliser cette structure a été fait pour optimiser la mémoire et éviter les problèmes de débordement pendant la lecture. Nous avons commencé d'abord par récupérer la taille de cette ligne que nous envoyons au client afin qu'il alloue une espace dans la mémoire de cette taille et ensuite on envoie la ligne extraite du fichier texte au client qui lui va formater les données à partir de la ligne reçue par une fonction d'affichage et affiche que les éléments demandés. La fonction d'affichage de cette requête est spéciale pour l'affichage du résultat de cette requête, et différente des autres fonctions d'affichage des autres requêtes.



5.2 Fonction de recherche :

Pour cette requête nous avons créé une méthode qui s'appelle **req_reference(int ref, char *filename)** qui prend en paramètre une référence et le nom d'un fichier. Après l'ouverture du fichier et la gestion de l'erreur cette fonction parcourt le fichier par une boucle while qui tant que la fonction **getline()**, « qui rend un entier correspondant à la taille de cette ligne et stocke la ligne suivante d'un fichier texte dans une variable prise en paramètre », est strictement supérieure de 1, prend la ligne prochaine et récupère le premier élément délimité par # avec la fonction **strtok()** qui prend la ligne et le délimiteur en paramètre, Ensuite sur chaque itération on compare la référence récupérée avec la référence donnée par l'utilisateur et si la comparaison est vraie on stocke dans line de la structure Ligne la ligne de cette itération sans oublier de stocker le résultat du **getline** dans la taille de la structure Ligne.

Si aucune référence n'est pas trouvée on stocke dans la structure une ligne d'erreur et sa taille que nous allons renvoyer au serveur par la suite.

```
typedef struct Ligne
{
    char *line;
    int taille;
} Ligne;
```

5.3 Résultats :

```
mesouaks@im2ag-turing:~/L3_MIAGE/Sys
o teme_et_reseaux/Groupe3]: ./server 96
72 livre.txt
socket d'ouverture ouvert
lison établie avec succès
service ouvert avec succès
A l'écoute ...
Le serveur attend une connexion...
Connexion établie sur le port : 9672,
Identificateur du nouveau socket: 4
Processus principale serveur attend
une connexion...
Le fils a été créé 595393
Connexion établie sur le port : 9672,
Identificateur du nouveau socket: 4
Processus principale serveur attend
une connexion...
Le fils a été créé 595495
Connexion établie sur le port : 9672,
Identificateur du nouveau socket: 4
Processus principale serveur attend
une connexion...
Le fils a été créé 595631
Mort du fils
[]

mesouaks@im2ag-turing:~/L3_MIAGE/Sys
o teme_et_reseaux/Groupe3]: ./client in
2ag-turing 9672
TCP server a été bien crée.
Le client est connecté au serveur.
-----
--MENU--
-----
Bienvenue dans le serveur
1->livre depuis sa reference
2->livre par mots clés :
3->Savoir la reference et le titre d
e puis l'auteur et le genre
4-> Tapez 5 pour sortir

Quel requête voulez vous le faire
(Taper 5 pour sortir)
choix requete : 1

    Quelle votre reference: 141

141Auteur : Victor Hugo

titre de : Notre-Dame de Paris

genre :roman

nombre de page : superieur à 300

Quel requête voulez vous le faire
(Taper 5 pour sortir)
choix requete : 5

Merci pour votre visite , a bientot
Le client a quitté le serveur .
mesouaks@im2ag-turing:~/L3_MIAGE/Sys
o teme_et_reseaux/Groupe3]: []

mesouaks@im2ag-turing:~/L3_MIAGE/Sys
o teme_et_reseaux/Groupe3]: ./client in
2ag-turing 9672
TCP server a été bien crée.
Le client est connecté au serveur.
-----
--MENU--
-----
Bienvenue dans le serveur
1->livre depuis sa reference
2->livre par mots clés :
3->Savoir la reference et le titre d
e puis l'auteur et le genre
4-> Tapez 5 pour sortir

Quel requête voulez vous le faire
(Taper 5 pour sortir)
choix requete : 1

    Quelle votre reference: 27

reference inexistante

Quel requête voulez vous le faire
(Taper 5 pour sortir)
choix requete : 1

    Quelle votre reference: 10

10Auteur : Jack London

titre de : Aventures en mer

genre :recits

nombre de page : superieur à 300

Quel requête voulez vous le faire
(Taper 5 pour sortir)
choix requete : []

mesouaks@im2ag-turing:~/L3_MIAGE/Sys
o teme_et_reseaux/Groupe3]: ./client in
2ag-turing 9672
TCP server a été bien crée.
Le client est connecté au serveur.
-----
--MENU--
-----
Bienvenue dans le serveur
1->livre depuis sa reference
2->livre par mots clés :
3->Savoir la reference et le titre d
e puis l'auteur et le genre
4-> Tapez 5 pour sortir

Quel requête voulez vous le faire
(Taper 5 pour sortir)
choix requete : 1

    Quelle votre reference: 96

reference inexistante

Quel requête voulez vous le faire
(Taper 5 pour sortir)
choix requete : 1

    Quelle votre reference: 140

140Auteur : ictor ugo

titre de : Les travailleurs de la me
r

genre :roman

nombre de page : superieur 300

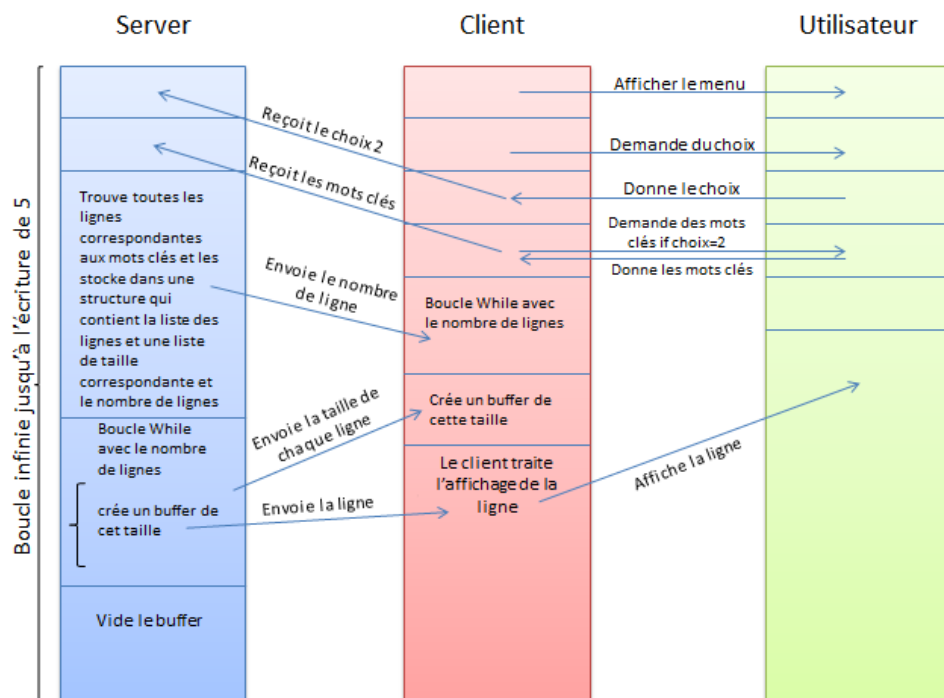
Quel requête voulez vous le faire
(Taper 5 pour sortir)
choix requete : []
```

6) Deuxième requête :

Le client soumet un ou des mots clés, et reçoit comme affichage les références, les noms des auteurs, les titres et les genres de toutes les livres triés par ordre alphabétique dont les titres contiennent ce ou ces mots clés. Si le client soumet un ou des mots clés qui n'existent dans aucun des titres des livres, on lui affiche un message qui indique que ces mots clés sont introuvables dans aucun des titres et on lui redemande de saisir un ou des mots existants *tant qu'il ne tape pas 5*.

6.1 Protocole d'échange de cette requête :

Après l'envoi du menu par le client, il attend le retour du choix 2, le client va envoyer ce choix au serveur qui va se mettre dans le case des mots clés. Ensuite il envoie les mots clés au serveur afin d'appeler les fonctions qui vont chercher dans le fichier texte les lignes qui contiennent les mots clés. L'envoi des mots clés du client au serveur se fait par une chaîne de caractère de taille 1000. Le serveur trouve toutes les lignes correspondantes aux mots clés et les stocke dans une structure qui contient la liste des lignes et une liste de taille correspondante et le nombre de lignes comme afficher dans le premier cas. La fonction d'affichage de cette requête est spéciale pour l'affichage du résultat de cette requête, et différente des autres fonctions d'affichage des autres requêtes.



6.2 Fonction de recherche :

La première fonction **ligneParMot** renvoie la ligne où le titre contient le mot. Après on passe à la deuxième fonction **tailleParMot**, qui renvoie la taille de la ligne où le titre contient le mot clé passer en paramètre. Après on fait appel à la fonction **keyWord_request** renvoie une structure qui contient une liste de ligne où on a stocké les lignes correspondantes aux mots clés et leur taille respective et le nombre de réponse trouvé.

Si un mot clé n'est pas trouvé on stocke dans la structure une ligne d'erreur et sa taille que nous allons renvoyer au serveur par la suite.

6.3 Résultats :

```
mesouaks@im2ag-turing: [~/L3_MIAGE/Sys-
tème_et_réseaux/Groupe3] : ./server 96
73 livre.txt
socket d'ouverture ouvert
liaison établie avec succès
service ouvert avec succès
A l'écoute ...
Le serveur attend une connexion...
Connexion établie sur le port : 9673,
Identificateur du nouveau socket: 4
Processus principale serveur attend
une connexion...
Le fils a été créé 597829
Connexion établie sur le port : 9673,
Identificateur du nouveau socket: 4
Processus principale serveur attend
une connexion...
Le fils a été créé 597881
Connexion établie sur le port : 9673,
Identificateur du nouveau socket: 4
Processus principale serveur attend
une connexion...
Le fils a été créé 597977
Mort du fils
[]

-----
Bienvenue dans le serveur
1->livre depuis sa référence
2->livre par mots clés :
3->savoir la référence et le titre d
e puis l'auteur et le genre
4-> Tapez 5 pour sortir

Quel requête voulez vous le faire
(Taper 5 pour sortir)
choix requete : 2

Tapez vos mots :Dame Anas

-----
reference : 141
Auteur : Victor Hugo
titre : Notre-Dame de Paris
genre : roman

-----
reference : aucun des mots clé n'exis
te
Auteur : (null)

Quel requête voulez vous le faire
(Taper 5 pour sortir)
choix requete : []

(Taper 5 pour sortir)
choix requete :
2

Tapez vos mots :Salah noir Walid

-----
reference : aucun des mots clé n'exis
te
Auteur : (null)

Quel requête voulez vous le faire
(Taper 5 pour sortir)
choix requete : 2

Tapez vos mots :Notre

-----
reference : 80
Auteur : Stendhal
titre : Le rouge et le noir
genre : roman

-----
reference : aucun des mots clé n'exis
te
Auteur : (null)

Quel requête voulez vous le faire
(Taper 5 pour sortir)
choix requete : 5

Merci pour votre visite , a bientôt

Le client a quitté le serveur .
mesouaks@im2ag-turing: [~/L3_MIAGE/Sy
stème_et_réseaux/Groupe3] :
TCP server a été bien créé.
Le client est connecté au serveur.
--MENU--
-----
Bienvenue dans le serveur
1->livre depuis sa référence
2->livre par mots clés :
3->savoir la référence et le titre
depuis l'auteur et le genre
4-> Tapez 5 pour sortir

Quel requête voulez vous le faire
(Taper 5 pour sortir)
choix requete : 2

Tapez vos mots :Notre

-----
reference : 141
Auteur : Victor Hugo
titre : Notre-Dame de Paris
genre : roman

-----
reference : aucun des mots clé n'exis
te
Auteur : (null)

Quel requête voulez vous le faire
(Taper 5 pour sortir)
choix requete : 5

Merci pour votre visite , a bientôt

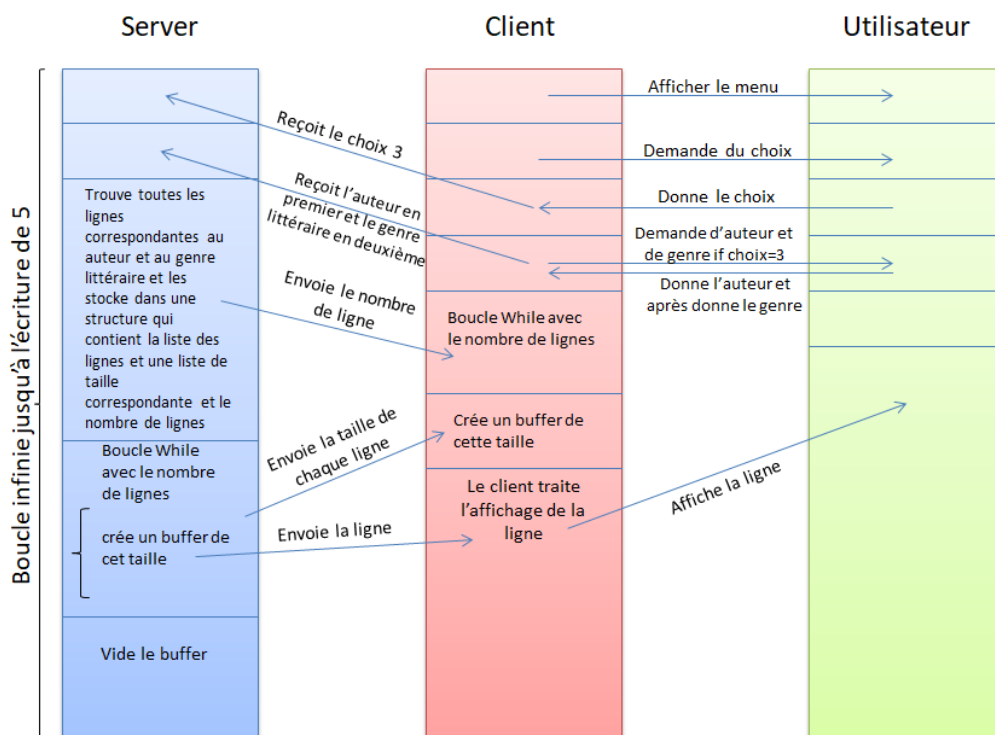
Le client a quitté le serveur .
mesouaks@im2ag-turing: [~/L3_MIAGE/Sy
stème_et_réseaux/Groupe3] :
```

7) Troisième requête :

Le client soumet le nom de l'auteur et le genre littéraire, et reçoit comme affichage les références, les titres, les nombres de pages des livres concernés. Si le client soumet un auteur qui n'est pas dans la liste ou un genre littéraire inexistant ou les deux, on lui affiche un message indiquant que les informations fournies sont introuvables dans la liste et on lui redemande de saisir un nom d'auteur et un genre littéraire qui existent *tant qu'il ne tape pas* 5.

7.1 Protocole d'échange de cette requête :

Après l'envoi du menu par le client, il attend le retour du choix 3, le client envoie ce choix au serveur ensuite ce dernier se place dans le case d'auteur et genre littéraire. Le client va demander à l'utilisateur l'auteur puis le genre littéraire. Ensuite, il envoie l'auteur puis le genre au serveur afin d'appeler la fonction qui trouve toutes les lignes correspondantes au auteur et au genre littéraire et les stocke dans une structure qui contient la liste des lignes et une liste de taille correspondante et le nombre de lignes. La fonction d'affichage de cette requête est spéciale pour l'affichage du résultat de cette requête, et différente des autres fonctions d'affichage des autres requêtes.



7.2 Fonction de la recherche :

Pour la 3ème requête on a défini une **fonction auteurGenreRequete(char auteur,chargenre, char* filename)** qui prend donc en paramètre une chaîne de caractère composée du nom et prénom de l'auteur recherché. En second paramètre le genre recherché et enfin le nom du fichier dans lequel on cherche les données. On parcourt ensuite notre fichier ligne par ligne en comparant le nom d'auteur voulu avec le nom d'auteur de la ligne actuellement lu. On fait de même en comparant le genre, on a au préalable récupéré le nom et le genre de chaque ligne grâce au fonction de découpe appliquée à notre ligne, respectivement **authorCheck()** et **genderCheck()**. Dans le cas où les 2 correspondent on

recupère la ligne brut et on l'envoie au client de façon similaire aux précédentes requêtes qui se chargera à son tour d'une découpe et un affichage approprié à la réponse attendue, c'est à dire une référence et un titre. En cas d'échec de réponse positive le serveur renvoie à la manière d'une réponse « aucun résultat trouve pour cette combinaison ».

7.3 Résultats :

<pre>mesouaks@im2ag-turing:~/L3_MIAGE/Sys o teme_et_reseaux/Groupe3]: ./server 96 73 livre.txt socket d'ouverture ouvert liason etablie avec succès service ouvert avec succès A l'écoute ... Le serveur attend une connection... Connexion établie sur le port : 9673, Identificateur du nouveau socket: 4 Processus principale serveur attend une connection... Le fils a été créé 599067 Connexion établie sur le port : 9673, Identificateur du nouveau socket: 4 Processus principale serveur attend une connection... Le fils a été créé 599117 Connexion établie sur le port : 9673, Identificateur du nouveau socket: 4 Processus principale serveur attend une connection... Le fils a été créé 599193 Mort du fils Mort du fils []</pre>	<pre>(Taper 5 pour sortir) choix requete : 3 Quel est auteur : Moliere Quel est le genre :theatre ----- Informations ----- reference : 20 titre : Dom Juan ----- Informations ----- reference : 110 titre : Les fourberies de Scapin ----- Informations ----- reference : 130 titre : Les precieuses ridicules ----- Informations ----- Quel requête voulez vous le faire (Taper 5 pour sortir) choix requete : []</pre>	<pre>mesouaks@im2ag-turing:~/L3_MIAGE/Sys o teme_et_reseaux/Groupe3]: ./client im 2ag-turing 9673 TCP server a été bien créée. Le client est connecté au serveur. --MENU-- ----- Bienvenue dans le serveur 1->livre depuis sa reference 2->livre par mots clés : 3->Savoir la reference et le titre d epuis l'auteur et le genre 4-> Tapez 5 pour sortir Quel requête voulez vous le faire (Taper 5 pour sortir) choix requete : 3 Quel est auteur : Mowad Quel est le genre :theatre aucun resultat trouve pour cette comb inaison Quel requête voulez vous le faire (Taper 5 pour sortir) choix requete : 5 Merci pour votre visite , a bientot Le client a quitté le serveur . mesouaks@im2ag-turing:~/L3_MIAGE/Sys o teme_et_reseaux/Groupe3]: []</pre>	<pre>mesouaks@im2ag-turing:~/L3_MIAGE/Sy o teme_et_reseaux/Groupe3]: ./client im2ag-turing 9673 TCP server a été bien créée. Le client est connecté au serveur. --MENU-- ----- Bienvenue dans le serveur 1->livre depuis sa reference 2->livre par mots clés : 3->Savoir la reference et le titre depuis l'auteur et le genre 4-> Tapez 5 pour sortir Quel requête voulez vous le faire (Taper 5 pour sortir) choix requete : 3 Quel est auteur : Walid Quel est le genre :Salah aucun resultat trouve pour cette com binaison Quel requête voulez vous le faire (Taper 5 pour sortir) choix requete : 5 Merci pour votre visite , a bientot Le client a quitté le serveur . mesouaks@im2ag-turing:~/L3_MIAGE/Sy o teme_et_reseaux/Groupe3]: []</pre>
--	--	---	---

8) Compilation et exécution:

Notre programme se compose de plusieurs fichiers sources, il faut pouvoir gérer les dépendances entre ces fichiers. Pour compiler notre programme, on va utiliser un Makefile. A l'aide de ce Makefile, on va pouvoir faire des éditions de lien.

Après avoir tapé la commande make dans le terminal, s'il n'y a pas d'erreurs les fichiers vont se compiler de façon normale comme dans notre cas.

```
• elkbabtm@im2ag-turing:~]: cd Groupe3_renduFinale
• elkbabtm@im2ag-turing:~/Groupe3_renduFinale]: ls
bin doc headers Makefile sources
• elkbabtm@im2ag-turing:~/Groupe3_renduFinale]: make clean
rm ./bin/*
• elkbabtm@im2ag-turing:~/Groupe3_renduFinale]: make
gcc -Wall -c ./sources/discuServer.c -o ./bin/discuServer.o
gcc -Wall -c ./sources/req.c -o ./bin/req.o
gcc -Wall -c ./sources/server.c -o ./bin/server.o
gcc -Wall ./bin/discuServer.o ./bin/req.o ./bin/server.o -o ./bin/server
gcc -Wall -c ./sources/client.c -o ./bin/client.o
gcc -Wall -c ./sources/discuClient.c -o ./bin/discuClient.o
gcc -Wall ./bin/client.o ./bin/req.o ./bin/discuClient.o -o ./bin/client
• elkbabtm@im2ag-turing:~/Groupe3_renduFinale]:
```

Et pour l'exécution :

```

o elkhbat@in2ag-turing:~/Groupe3_renduFinale$ bin/server 9667 doc/livre.txt
socket d'ouverture ouvert
liaison établie avec succès
service ouvert avec succès
A l'écoute ...
Le serveur attend une connexion...
Connexion établie sur le port : 9667, Identificateur du nouveau socket: 4
Processus principale serveur attend une connexion...
Le fils a été créé 731621
[ ]

o elkhbat@in2ag-turing:~/Groupe3_renduFinale$ bin/client in2ag-turing 9667
TCP server a été bien créé.
Le client est connecté au serveur.
-----MENU-----
Bienvenue dans le serveur
1->livre depuis sa reference
2->livre par mots clés (ils doivent être dans le titre )
3->Savoir la reference et le titre depuis l'auteur et le genre
4-> Tapez 5 pour sortir

Quel requête voulez vous le faire
(Taper 5 pour sortir)
choix requete : [ ]

```

Dans le cas où on écrit un chemin invalide du livre, on aura comme résultat :

```

o elkhbat@in2ag-turing:~/Groupe3_renduFinale$ cd Groupe3_renduFinale
o elkhbat@in2ag-turing:~/Groupe3_renduFinale$ ls
bin doc headers Makefile sources
o elkhbat@in2ag-turing:~/Groupe3_renduFinale$ make clean
rm ./bin/*
o elkhbat@in2ag-turing:~/Groupe3_renduFinale$ make
gcc -Wall -c ./sources/discuServer.c -o ./bin/discuServer.o
gcc -Wall -c ./sources/req.c -o ./bin/req.o
gcc -Wall -c ./sources/server.c -o ./bin/server.o
gcc -Wall ./bin/discuServer.o ./bin/req.o ./bin/server.o -o ./bin/server
gcc -Wall -c ./sources/client.c -o ./bin/client.o
gcc -Wall -c ./sources/discuClient.c -o ./bin/discuClient.o
gcc -Wall ./bin/client.o ./bin/req.o ./bin/discuClient.o -o ./bin/client
o elkhbat@in2ag-turing:~/Groupe3_renduFinale$ bin/server 9669 livre.txt
socket d'ouverture ouvert
liaison établie avec succès
service ouvert avec succès
A l'écoute ...
Le serveur attend une connexion...
Connexion établie sur le port : 9669, Identificateur du nouveau socket: 4
Processus principale serveur attend une connexion...
Le fils a été créé 726917
fichier inexistant
Mort du fils
[ ]

o elkhbat@in2ag-turing:~/Groupe3_renduFinale$ bin/client in2ag-turing 9669
TCP server a été bien créé.
Le client est connecté au serveur.
-----MENU-----
Bienvenue dans le serveur
1->livre depuis sa reference
2->livre par mots clés (ils doivent être dans le titre )
3->Savoir la reference et le titre depuis l'auteur et le genre
4-> Tapez 5 pour sortir

Quel requête voulez vous le faire
(Taper 5 pour sortir)
choix requete : 1

    Quelle votre reference: 10

(null)
Quel requête voulez vous le faire
(Taper 5 pour sortir)
choix requete : [ ]

```

9) Limitations:

1 - Au début, on avait une fonction qui fait la requête 1 et 2, et fait l'affichage en même temps, et donc le serveur envoie les réponses traitées et formates, et le client fait qu'un printf de la réponse.

2- On a su que le client doit avoir le rôle de formater les réponses, et donc on a refait une fonction de formatage pour chaque requête.

3- On n'avait plus le temps d'optimiser les fonctions d'affichage car ils font presque la même chose, et pour cela on a 3 fonctions de recherche, une pour chaque requête.

4- On n'a pas pensé à régler le problème de la taille jusqu'au mercredi après-midi, et donc on a résolu le problème et on a posé le protocole expliqué en haut.

5- On n'avait pas du temps pour terminer la requête 4, il bugait nos serveur/client et donc on ne l'a pas ajouté dans notre programme.

6- Pour l'envoi des mots clés du client au serveur, on a utilisé une chaîne de caractère qui dépasse pas 1000, et donc il ne faut pas entrés plus que 1000 char, Même chose pour l'envoi de l'auteur et le genre

7- Pour l'auteur et le genre, les noms doivent être les mêmes que dans le fichier, et donc si il y avait une majuscule, l'utilisateur doit la respecter sinon on ne renvoie pas la réponse, exemple : Victor Hugo renvoie auteur introuvables.

8- la fonction `key_request()` affiche que 5 mots clés, donc même si on entre 10 mots clés, la fonction va afficher que 5 mots, et elle va afficher que les autres mots sont introuvables. On a essayé d'augmenter les 5 mots à 20 même à cause du temps on n'a pas pu avancer et donc on se limite à 5 mots.

9-On n'a passé aussi du temps pour le faire le Makefile et traité les fichier.o et les mettre dans l'arborescence donné, on est arrivé à le faire au dernier moment.

10) Conclusion:

Tout d'abord, ce projet nous a permis d'appliquer les connaissances que nous avons acquises durant ce semestre en Systèmes/Réseaux et nous a donné une occasion bénéfique sur différents plans technique et technologique. Il est vrai que créer une application et respecter des consignes rend un projet intéressant et professionnel mais il y a aussi toutes les démarches qui ne sont pas visibles et qui rendent enrichissante une telle expérience.

Cette initiative n'est qu'un début pour plusieurs autres projets beaucoup plus étendus. Nous espérons que nous étions à la hauteur et que notre effort sera apprécié.