

Hatim LAGHRISSI

Ayman SALOUH

Souleymen OUCHANE

Groupe 06



Projet Système/Réseaux

Décembre 2022

Read me :

Après notre passage à l'oral, nous avons pu continuer le code et améliorer certains points du projet.

Nous avons pu finir la **requête 3** qui consistait à demander à l'utilisateur un auteur et renvoyer un résultat selon un critère si plusieurs livres étaient trouvés avec l'auteur saisi. Le critère qui n'était pas encore fonctionnel était celui concernant le renvoi du premier livre avec la meilleure appréciation. Il est maintenant implémenté dans le code correctement.

Nous avons aussi réussi à debugger le **requête 2** qui n'était pas du tout fonctionnel lors de l'oral. Il consiste à demander à l'utilisateur un auteur et un genre. On reçoit en retour la référence et le titre des livres concernant cette recherche.

On a aussi pu **minimiser le code** en utilisant des fonctions dans les switch requête du main_server.c et client.c afin d'avoir un code plus lisible.

Le **makefile** permettait de créer directement les exécutables sans passer par les étapes de création d'objets `.o`. Bien entendu, nous avons réglé cela, il permet maintenant de créer les objets .o avant de créer les exécutables. Les objets `.o` sont effacés automatiquement après la création des exécutables.

On avait un problème lorsque que l'utilisateur rentrait une chaîne de caractère au lieu d'un entier qui nous créait une boucle infinie dans notre processus. On a réglé ça en demandant une chaîne de caractère à la place d'un int. On transforme ensuite

celle-ci en int à l'aide de la fonction `atoi`. On a juste laissé ce bug pour la saisie des requêtes au tout début parce qu'avec `atoi`, on aurait dû enlever la valeur saisie à 0. Cela posait problème car nous devons modifier la valeur de nos requêtes. Donc pour éviter toute confusion dans notre code, nous avons laissé comme ça pour les requêtes.

Introduction:

Ce **projet consiste** à réaliser en C une application client/serveur TCP/IP simplifiée, cela consistera à la réalisation, **d'un côté** *un serveur* qui va interroger un fonds documentaire (fichier texte qui simule une base donnée) contenant des informations sur des livres sous le format (référence#auteur# titre#genre#nombre de pages#appréciation), **de l'autre** côté *un client* qui va pouvoir échanger avec *l'utilisateur* pour récupérer ses demandes et par la suite échanger avec le serveur via le protocole TCP/IP pour récupérer les données nécessaires et ensuite les afficher à l'utilisateur.

La **réalisation attendue** est d'écrire **en premier** lieu un *noyau client/serveur* c'est-à-dire que tout doit être en place pour que l'envoi d'une donnée X entre machine client et machine serveur puisse être possible. **En second** lieu, on rentre dans une phase plus conceptuel où on réfléchit sur papier le *protocole d'échange* entre client et serveur (le défi est qu'en face de chaque `write()` il doit y avoir un `read()` d'un même nombre d'octets), et on décide quels *types de donnée* va transiter. **Enfin**, on *implémente* notre code et on le teste à chaque fois.

Ce projet permet d'illustrer des notions réseaux tels que les protocoles de la couche 4 de Transport (TCP utilisé) (UDP non utilisé) et de voir le côté applicatif pour les utiliser en langage C, et de manipuler les notions de port, mais aussi de reprendre quelques notions système tels que la création de processus, leurs déléguer des algorithmes, la gestion de leur mort, `read` et `write` sur des descripteur de fichier, `makefile`, allocation de la mémoire (heap), programmation modulaire...

Pourquoi le protocole TCP et pas UDP ?

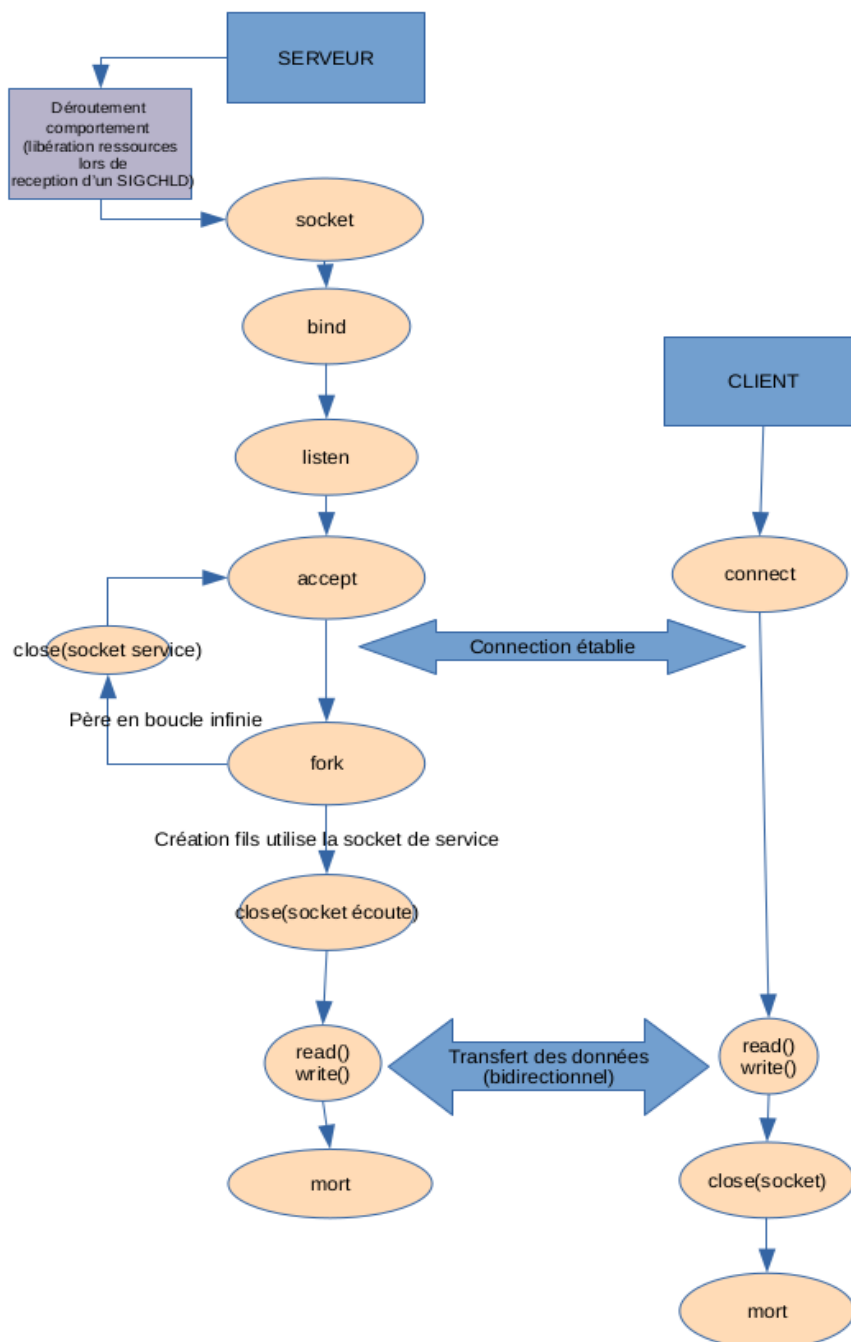
On programme une application client/serveur TCP/IP. On utilise le protocole TCP qui requiert d'**établir une connexion** pour pouvoir ensuite communiquer entre les deux machines et envoyer des paquets dans les deux sens à l'inverse du protocole UDP qui n'est pas une solution valable pour l'application client/serveur car ne requiert pas un établissement de connexion. La communication se fait d'une machine vers une machine cible, et on vérifie pas si les données ont bien été reçues, ce dernier est plus utilisé pour les transmissions où le facteur temps est primordial et si un datagramme manque cela n'impacte pas en grande partie l'intégrité du message (exemple jeux en ligne, appel audio). Tandis que le protocole TCP est utilisé là où le message transmis est important. Si un paquet manque, l'intégrité du message est compromise (exemple transfert fichier).

Le protocole TCP est adapté à la réalisation de l'application client/serveur car on a besoin de cet aspect d'échange qu'il offre, et on doit être sûr que l'intégralité des paquets soit délivrée.

Conception logiciel Global

Noyau

Schéma illustratif de la conception du noyau



Type de données transitant entre client et serveur :

Pour pouvoir établir un échange entre le client et le serveur, on a implémenté la procédure **dispatch**(*int* p, *void* *buffer, *int* taille) qui vérifie si les envois ont bien été fait et qu'il n'y a pas d'erreur et la procédure **receive**(*int* p, *void* *buffer, *int* taille) qui vérifie si tout a été lu et qu'il n'y a pas d'erreur.

Après réflexion, on a choisi d'utiliser des structures de données qui sont connues par le client et le serveur, ces structures ont des tailles prédéfinies et sont déclarées dans le serveur et le client. Une structure pour chaque type de requête correspond à ce que l'utilisateur devrait avoir comme résultat.

Struct reply0 {*char* auteur[MAX_AUTEUR], *char* titre[MAX_TITRE], *char* genre[MAX_GENRE], *int* nb_pages}

Struct reply1 {*int* reference, *char* auteur, *char* titre[MAX_TITRE], *char* genre[MAX_GENRE]}

Struct reply2 {*int* reference, *char* titre[MAX_TITRE]}

Struct reply3 {*char* auteur[MAX_AUTEUR], *char* titre[MAX_TITRE], *char* genre[MAX_GENRE], *int* nb_pages}

L'avantage avec ces structures est qu'on n'est pas obligé d'envoyer la taille avant d'envoyer l'information puisqu'on dispose des tailles à envoyer (**taille des structures**, **int** pour les choix dans les menus, **chaîne de caractères** pour l'envoi des mots clés ...)

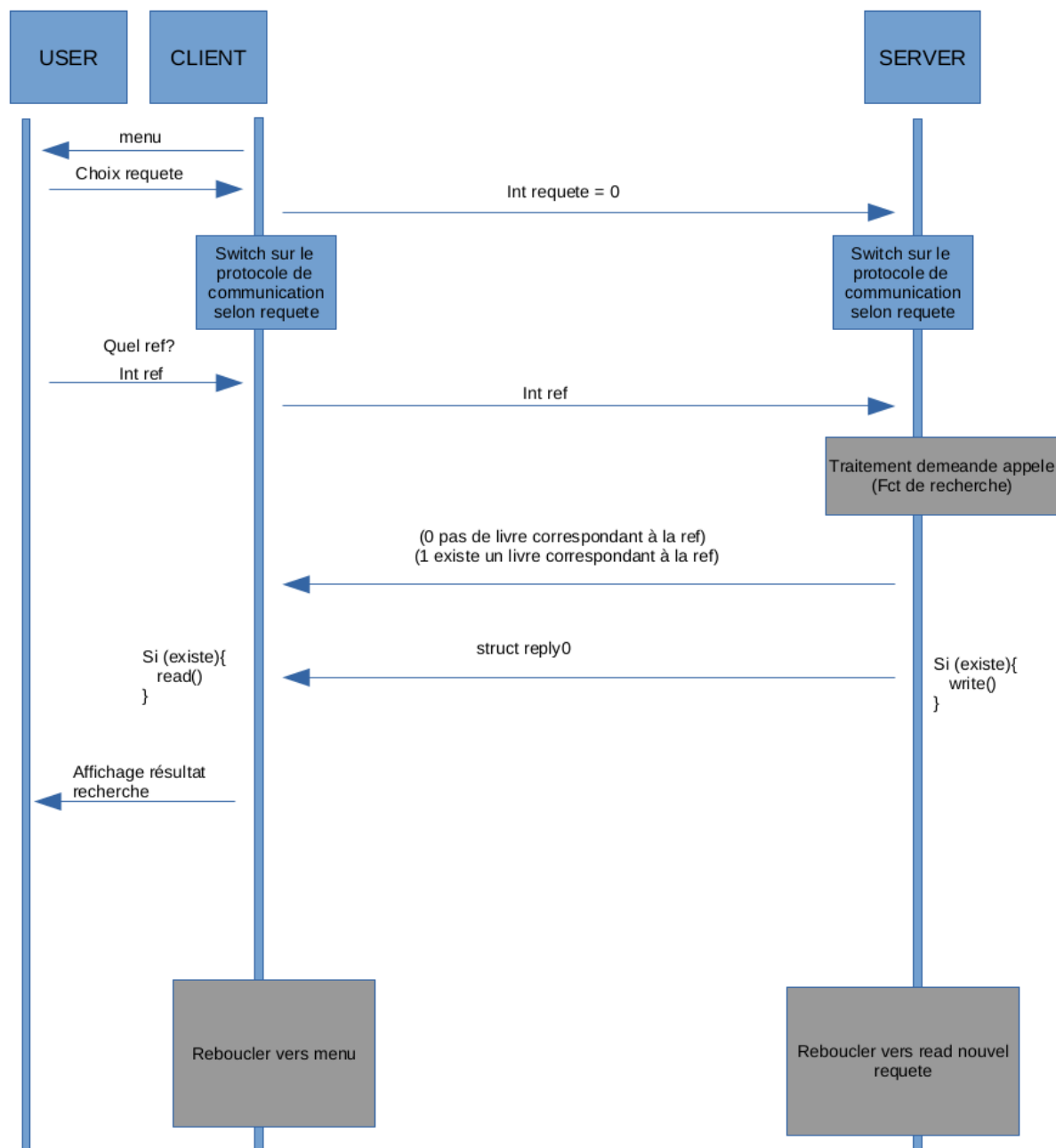
On a opté pour parser le fichier *lib.txt* en un tableau de structure

libresBD[taillesBD] contenant tous les livres de notre fichier.

La création du tableau est faite dans le fils pour qu'on puisse travailler dessus s'il y aura des mises à jour du fichier *lib.txt*.

Protocole d'échange client serveur

Protocole 0:

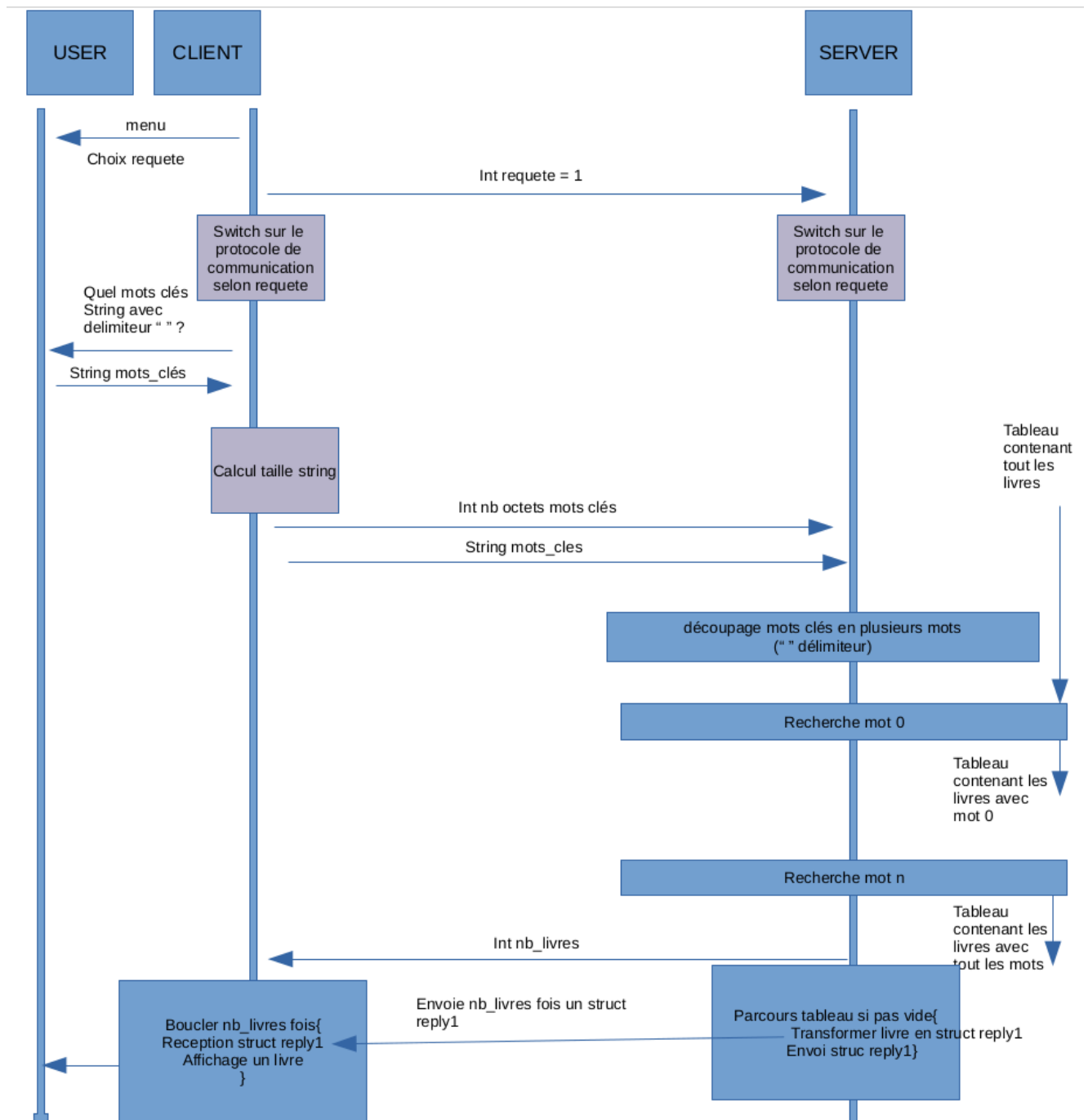


Le protocole de communication 0 est utilisé quand l'utilisateur souhaite soumettre une référence et avoir en retour si cette dernière existe les informations suivantes (nom auteur, titre livre, genre, nb pages). L'échange se déroule comme suit :

Le client affiche le menu à l'utilisateur qui répond en saisissant un entier entre $[-1, 3]$, dans ce cas il choisit le 0, cette information est envoyée au serveur sous forme d'entier, ensuite le client et le serveur switch sur cet entier pour rentrer en communication synchrone (càd **read et write au même niveau avec le même nombre d'octets**). Le client demande à l'utilisateur quelle référence il souhaite chercher. Cette information est envoyée au serveur, qui appelle

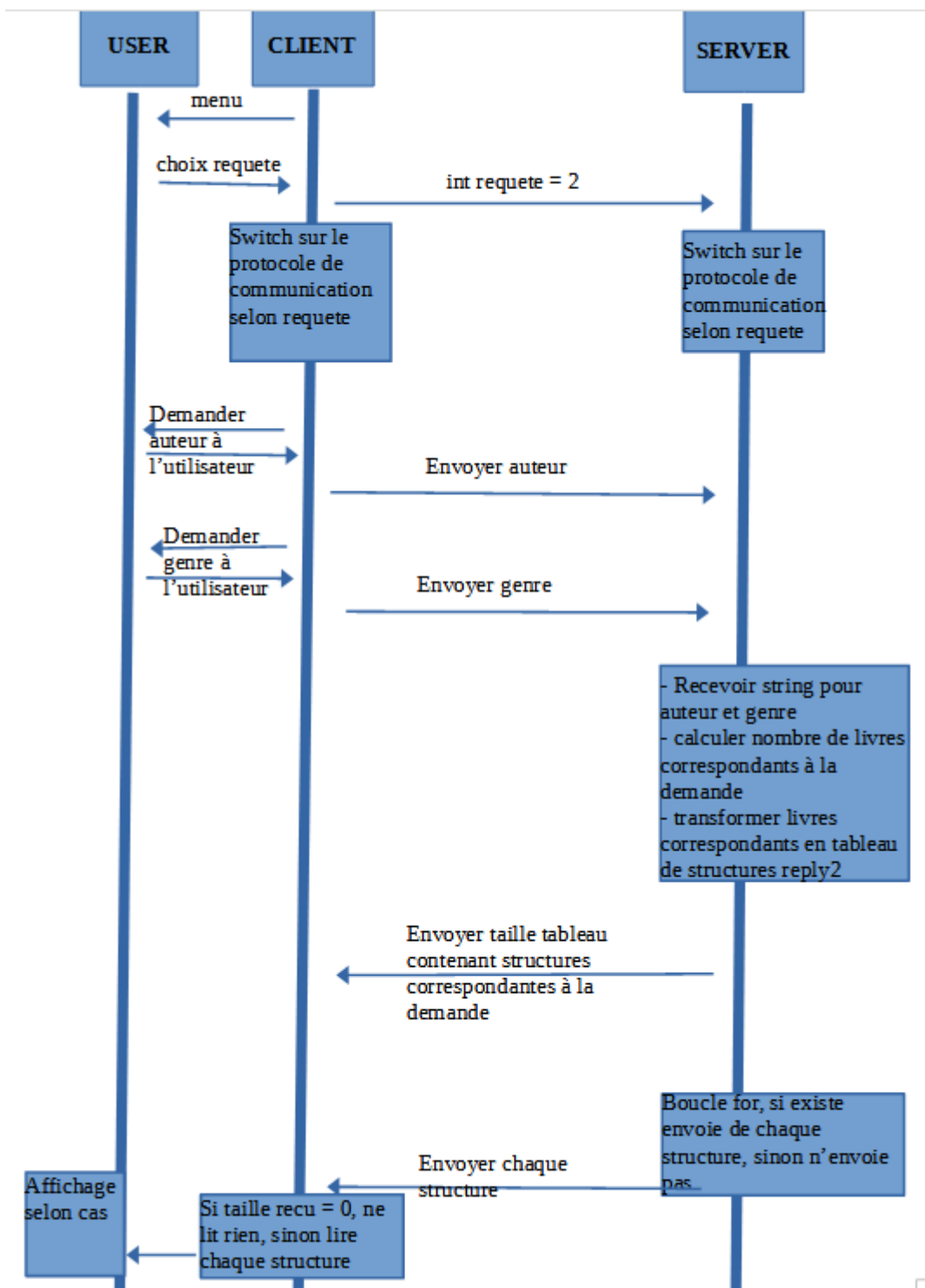
la fonction recherche ref, à cette étape on sait si la référence existe ou pas. Cette information est renvoyée au client sous forme entière avec une valeur binaire (0 = ref n'existe pas) ou (1 = existe ref). Si la référence existe le serveur envoie la structure reply 0 au client, cette dernière contient les informations souhaitées par le client, sa taille est connue dans les deux programmes donc il n'y a pas besoin d'envoyer la taille.

Protocole 1:



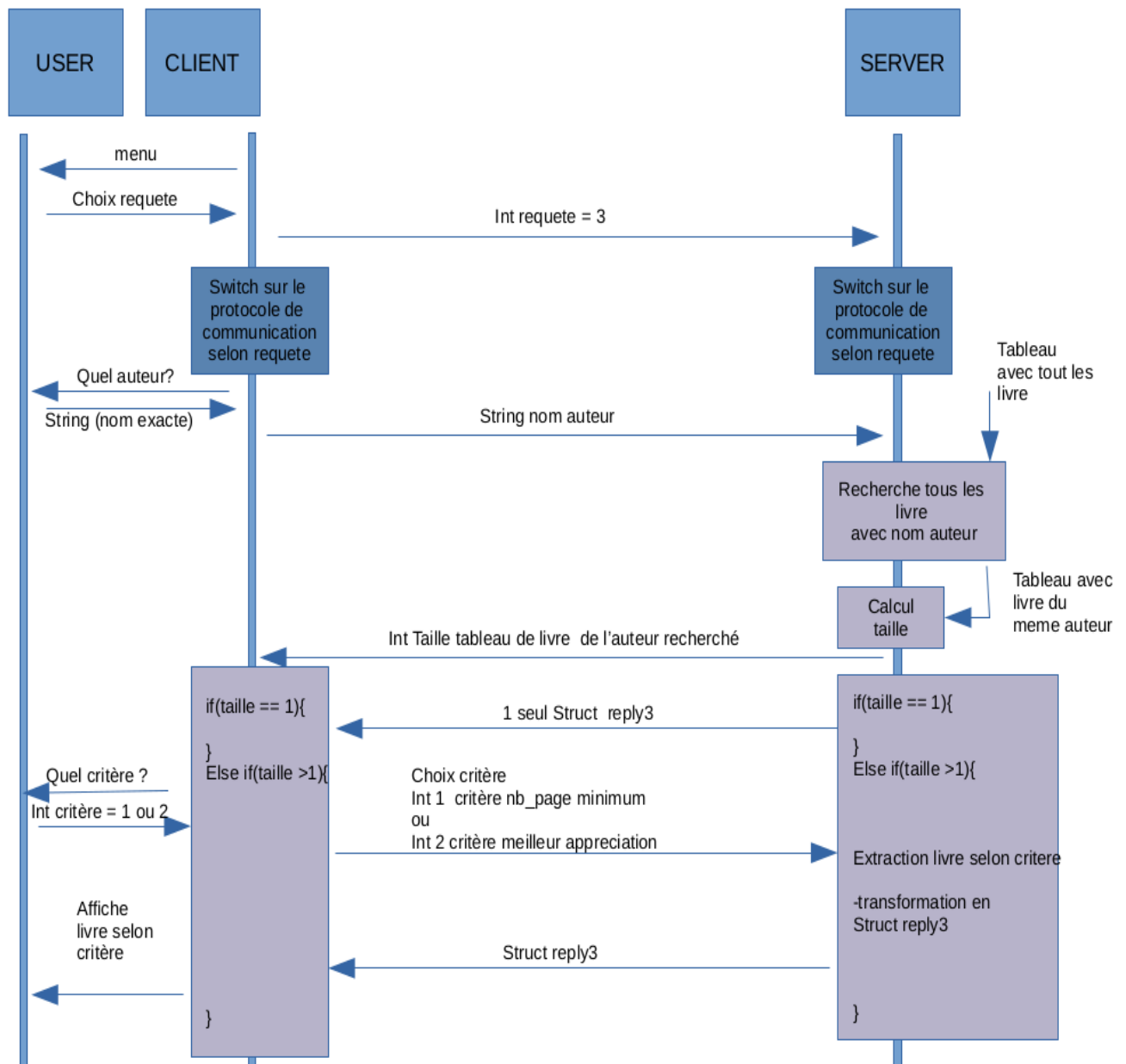
Le mot clé doit être de format : une chaîne de caractères séparée par des espaces. Le client calcule la taille et envoie la chaîne de caractères au serveur, ce dernier fait des traitements sur la chaîne de caractères (découpage, recherche des mots), il parcourt le tableau et le transforme en **struct reply1** et envoie tout cela au client qui fait l'affichage.

Protocole 2 :



Ce protocole consiste à récupérer le choix de l'utilisateur par le client, envoie le numéro au serveur pour qu'ils se mettent tous les deux dans le cas correspondant au choix. Ensuite, on récupère le nom complet de l'auteur et le genre dans deux chaînes de caractères différentes, on envoie les deux chaînes via la procédure **dispatch** et on connaît la taille max à envoyer. Le serveur reçoit les deux chaînes de caractères via la procédure **receive**, ensuite on calcule le nombre de livres correspondant aux choix (il faut que le nom d'auteur et le genre appartiennent au même livre). Quand on aura la taille, on fait une fonction qui met la référence et le titre dans un tableau de type **struct reply2**, on envoie la taille (si elle égale à 0 on envoie rien et on lit rien), ensuite on boucle avec **for** pour envoyer les **structure reply2** une par une, et on affiche le résultat selon le cas (existe ou pas).

protocole 3:

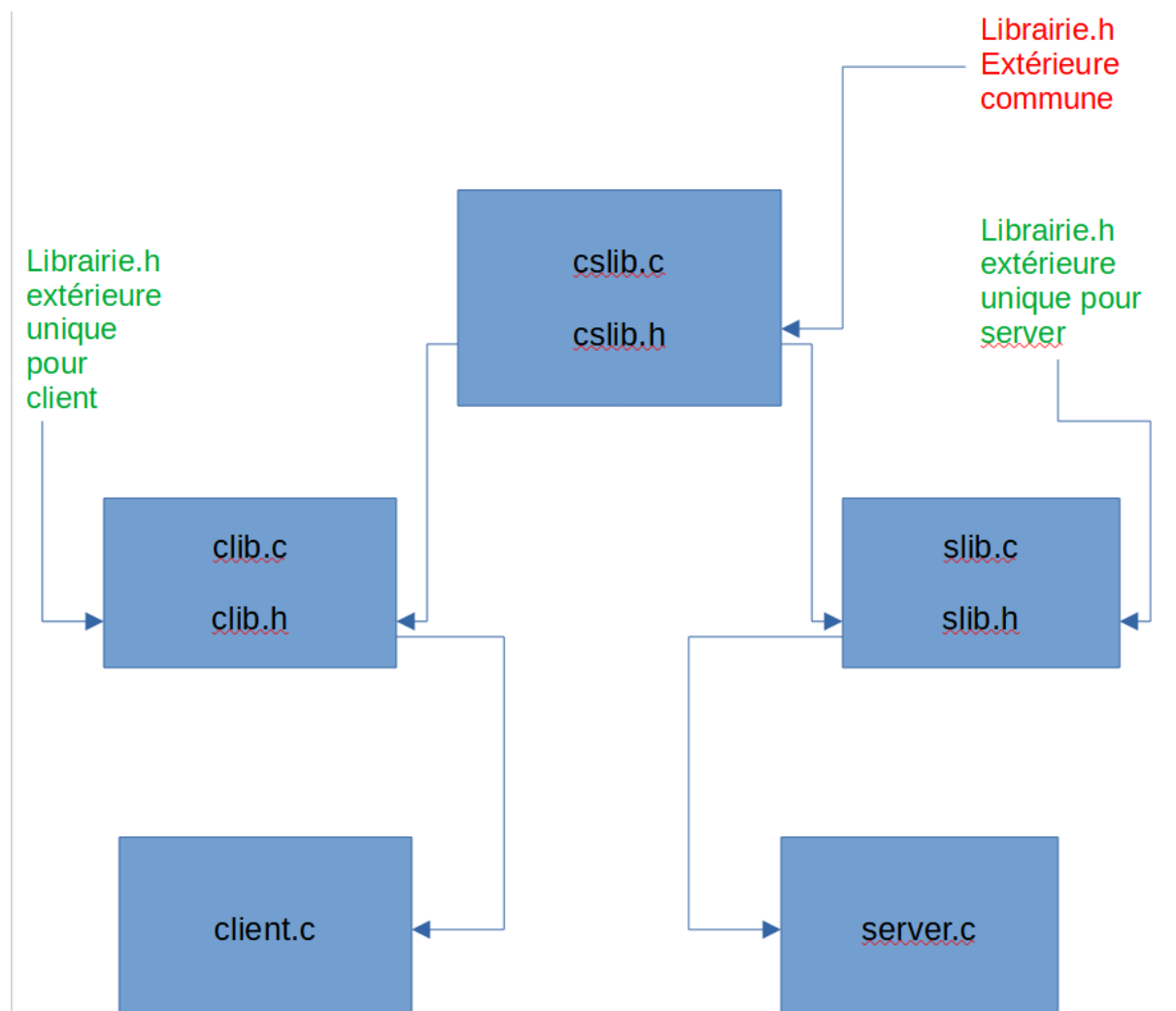


Ce protocole permet à l'utilisateur dans le cas où il tape 3 dans le menu de pouvoir envoyer au client un nom d'auteur complet et chercher selon un critère (1 pour **nb_page** ou 2 pour **meilleur appréciation**) le critère à utiliser, le serveur recherche les livres avec ce nom d'auteur et calcule la taille et envoie la taille, il reçoit le choix et fait l'extraction selon le critère et transforme en **struct reply3**, envoie cette structure au client pour afficher le résultat.

programmation modulaire

on a décidé de répartir nos fonctions sur plusieurs modules comme suit:

- **cslib.c** : contient le code source des fonctions utilisé dans les deux main() client.c et serveur.c
- **clib.c** : contient le code source des fonctions utilisé dans client .c mais pas dans server.c
- **slib.c** : contient le code source des fonctions utilisé dans server.c mais pas dans client.c
- **cslib.h** : contient les headers et aussi les structures de réponse (reply0, reply1, reply2, reply3), donc ils sont connu par le client et le serveur, dans ce fichier il ' y a aussi la la taille maximale des champs des structures reply.



l'illustration ci-dessus représente:

- les modules (rectangle bleu).
- les librairies standard utilisées (texte en rouge).
- les inclusions des headers (flèches) càd les includes "x.h" ou <x.h>

makefile

le makefile est créé de façon à respecter la programmation modulaire, donc on passe d'abord par la phase (préprocesseur, compilation, assemblage) donc on fait une **compilation séparé** et on génère des fichiers objets .o pour chaque module, l'**option -I de gcc** a été utilisé pour indiquer le chemin relatif des headers.

puis on passe à **la phase d'édition des liens** en générant un ou deux exécutable exécutables, cela dépend des arguments du programme **make**.

Expérimentations :

Nous allons illustrer chaque requête en montrant les réponses du client pour les recherches.

Requête 0 :

Requête demandant une référence et renvoie les informations du livre associé.

Cas où la référence est existante :

```
-----MENU-----
0 pour avoir (nom auteur, titre livre, genre, nb_pages) a partir de la reference.
1 pour avoir (reference, nom auteur, titre livre, genre) a partir des mot-clés.
2 pour avoir (reference, titre livre) a partir d'un auteur et d'un genre.
3 pour avoir (reference, titre livre, genre, nb_pages) a partir d'un auteur.
-1 pour quitter.
0
Quel reference desirez vous chercher ( entier naturel)?
10
RESULTAT DE LA RECHERCHE:
-----
Auteur = Jack London
Titre = Aventures en mer
Genre = recits
Plus de 300 pages.
```

Cas où la référence n'est pas trouvée :

```

-----MENU-----
0 pour avoir (nom auteur, titre livre, genre, nb_pages) a partir de la reference.
1 pour avoir (reference, nom auteur, titre livre, genre) a partir des mot-clés.
2 pour avoir (reference, titre livre) a partir d'un auteur et d'un genre.
3 pour avoir (reference, titre livre, genre, nb_pages) a partir d'un auteur.
-1 pour quitter.
0
Quel reference desirez vous chercher ( entier naturel)?
999999999
RESULTAT DE LA RECHERCHE:
-----
Reference n'existe pas.

```

Requête 1 :

Requête demandant des mots clés et renvoie les informations des livres ayant tous les mots clés dans le titre.

Cas pour un mot clé et livres trouvés :

```

-----MENU-----
0 pour avoir (nom auteur, titre livre, genre, nb_pages) a partir de la reference.
1 pour avoir (reference, nom auteur, titre livre, genre) a partir des mot-clés.
2 pour avoir (reference, titre livre) a partir d'un auteur et d'un genre.
3 pour avoir (reference, titre livre, genre, nb_pages) a partir d'un auteur.
-1 pour quitter.
1
Veuillez saisir le(s) mot(s) clé(s) recherché(s) dans le titre ?
(Saisir de la façon suivante : mot_clé1 mot_clé2 mot_clé3 ...)
mer
RESULTAT DE LA RECHERCHE:
-----
Résultat 0 :
-----
Référence = 10
Auteur = Jack London
Titre = Aventures en mer
Genre = recits
-----
Résultat 1 :
-----
Référence = 90
Auteur = Ernest Hemingway
Titre = Le vieil homme et la mer
Genre = roman
-----
Résultat 2 :
-----
Référence = 140
Auteur = Victor Hugo
Titre = Les travailleurs de la mer
Genre = roman
-----
Résultat 3 :
-----
Référence = 150
Auteur = Honore de Balzac
Titre = Un drame au bord de la mer
Genre = nouvelle
-----

```

Cas pour plusieurs mots clés et livres trouvés :

```

-----MENU-----
0 pour avoir (nom auteur, titre livre, genre, nb_pages) a partir de la reference.
1 pour avoir (reference, nom auteur, titre livre, genre) a partir des mot-clés.
2 pour avoir (reference, titre livre) a partir d'un auteur et d'un genre.
3 pour avoir (reference, titre livre, genre, nb_pages) a partir d'un auteur.
-1 pour quitter.
1
Veuillez saisir le(s) mot(s) clé(s) recherché(s) dans le titre ?
(Saisir de la façon suivante : mot_clé1 mot_clé2 mot_clé3 ...)
homme mer
RESULTAT DE LA RECHERCHE:
-----
Résultat 0 :
-----
Référence = 90
Auteur = Ernest Hemingway
Titre = Le vieil homme et la mer
Genre = roman
-----

```

Cas où la recherche ne trouve aucun résultat :

```

-----MENU-----
0 pour avoir (nom auteur, titre livre, genre, nb_pages) a partir de la reference.
1 pour avoir (reference, nom auteur, titre livre, genre) a partir des mot-clés.
2 pour avoir (reference, titre livre) a partir d'un auteur et d'un genre.
3 pour avoir (reference, titre livre, genre, nb_pages) a partir d'un auteur.
-1 pour quitter.
1
Veuillez saisir le(s) mot(s) clé(s) recherché(s) dans le titre ?
(Saisir de la façon suivante : mot_clé1 mot_clé2 mot_clé3 ...)
homme mer espace
Pas de résultat pour votre recherche.

```

Requête 2 :

Requête demandant un auteur et un genre et renvoie les informations des livres associés.

Cas où un livre est trouvé :

```

-----MENU-----
0 pour avoir (nom auteur, titre livre, genre, nb_pages) a partir de la reference.
1 pour avoir (reference, nom auteur, titre livre, genre) a partir des mot-clés.
2 pour avoir (reference, titre livre) a partir d'un auteur et d'un genre.
3 pour avoir (reference, titre livre, genre, nb_pages) a partir d'un auteur.
-1 pour quitter.
2
Quel nom d'auteur voulez-vous chercher ?
Souleyman
Quel genre voulez-vous chercher pour cet auteur ?
roman
RESULTAT DE LA RECHERCHE:
-----
Resultat 0 :
-----
Reference = 80
Titre = Le rouge et le noir
-----

```

Cas où des livres sont trouvés :

```
-----MENU-----
0 pour avoir (nom auteur, titre livre, genre, nb_pages) a partir de la reference.
1 pour avoir (reference, nom auteur, titre livre, genre) a partir des mot-clés.
2 pour avoir (reference, titre livre) a partir d'un auteur et d'un genre.
3 pour avoir (reference, titre livre, genre, nb_pages) a partir d'un auteur.
-1 pour quitter.
2
Quel nom d'auteur voulez-vous chercher ?
Victor Hugo
Quel genre voulez-vous chercher pour cet auteur ?
poesie
RESULTAT DE LA RECHERCHE:
-----
Resultat 0 :
-----
Reference = 40
Titre = La legende des siecles
-----
Resultat 1 :
-----
Reference = 100
Titre = Les contemplations
-----
Resultat 2 :
-----
Reference = 142
Titre = Odes et ballades
-----
```

Cas où aucun livre ne correspond à la recherche :

```
-----MENU-----
0 pour avoir (nom auteur, titre livre, genre, nb_pages) a partir de la reference.
1 pour avoir (reference, nom auteur, titre livre, genre) a partir des mot-clés.
2 pour avoir (reference, titre livre) a partir d'un auteur et d'un genre.
3 pour avoir (reference, titre livre, genre, nb_pages) a partir d'un auteur.
-1 pour quitter.
2
Quel nom d'auteur voulez-vous chercher ?
Victor Hugo
Quel genre voulez-vous chercher pour cet auteur ?
theatre
RESULTAT DE LA RECHERCHE:
-----
Aucun livre ne correspond à vos choix.
```

Requête 3 :

Requête demandant un auteur et renvoie les informations des livres associés selon un critère si plusieurs résultats.

Cas où un résultat trouvé :

```

-----MENU-----
0 pour avoir (nom auteur, titre livre, genre, nb_pages) a partir de la reference.
1 pour avoir (reference, nom auteur, titre livre, genre) a partir des mot-clés.
2 pour avoir (reference, titre livre) a partir d'un auteur et d'un genre.
3 pour avoir (reference, titre livre, genre, nb_pages) a partir d'un auteur.
-1 pour quitter.
3
Veuillez saisir un nom d'auteur :
(en respectant la casse cad la majuscule au début)
Souleyman
RESULTAT DE LA RECHERCHE:
-----
Résultat :
-----
Référence = 80
Titre = Le rouge et le noir
Genre = roman
Nombre de pages = 640
Appréciation = A
-----

```

Cas où plusieurs résultats et envoie d'un seul livre avec le critère page minimum :

```

-----MENU-----
0 pour avoir (nom auteur, titre livre, genre, nb_pages) a partir de la reference.
1 pour avoir (reference, nom auteur, titre livre, genre) a partir des mot-clés.
2 pour avoir (reference, titre livre) a partir d'un auteur et d'un genre.
3 pour avoir (reference, titre livre, genre, nb_pages) a partir d'un auteur.
-1 pour quitter.
3
Veuillez saisir un nom d'auteur :
(en respectant la casse cad la majuscule au début)
Victor Hugo
Plusieurs résultats trouvés !
Veuillez choisir le critère de recherche parmi les suivants :
1 - le nombre de pages minimum
2 - la meilleure appréciation des lecteurs
1
RESULTAT DE LA RECHERCHE:
-----
Résultat :
-----
Référence = 143
Titre = Quatre-vingt-treize
Genre = roman
Nombre de pages = 122
Appréciation = B
-----

```

Cas où plusieurs résultats et envoie d'un seul livre avec le critère meilleure appréciation :

```

-----MENU-----
0 pour avoir (nom auteur, titre livre, genre, nb_pages) a partir de la reference.
1 pour avoir (reference, nom auteur, titre livre, genre) a partir des mot-clés.
2 pour avoir (reference, titre livre) a partir d'un auteur et d'un genre.
3 pour avoir (reference, titre livre, genre, nb_pages) a partir d'un auteur.
-1 pour quitter.
3
Veuillez saisir un nom d'auteur :
(en respectant la casse cad la majuscule au début)
Victor Hugo
Plusieurs résultats trouvés !
Veuillez choisir le critère de recherche parmi les suivants :
1 - le nombre de pages minimum
2 - la meilleure appréciation des lecteurs
2
RESULTAT DE LA RECHERCHE:
-----
Résultat :
-----
Référence = 100
Titre = Les contemplations
Genre = poesie
Nombre de pages = 608
Appréciation = A
-----

```

Cas où aucun livre ne correspond à la recherche :

```

-----MENU-----
0 pour avoir (nom auteur, titre livre, genre, nb_pages) a partir de la reference.
1 pour avoir (reference, nom auteur, titre livre, genre) a partir des mot-clés.
2 pour avoir (reference, titre livre) a partir d'un auteur et d'un genre.
3 pour avoir (reference, titre livre, genre, nb_pages) a partir d'un auteur.
-1 pour quitter.
3
Veuillez saisir un nom d'auteur :
(en respectant la casse cad la majuscule au début)
Ayman
Aucun résultat trouvé.

```

Utilisation de l'application :

Dans cette partie, nous allons expliquer comment utiliser l'application. Nous commençons par la compilation du programme. Lors de cette explication, on dit que toutes les commandes sont saisies depuis le dossier projet c'est-à-dire au même endroit que le makefile.

Commandes de compilation :

- création de l'exécutable server :
\$ **make bin/server**
- création de l'exécutable client :
\$ **make bin/client**

- création des 2 exécutables en même temps :
\$ make
- suppression des exécutables (les fichiers .o sont effacés directement après la création des exécutables) :
\$ make clean

Commandes d'exécution :

- exécution server :
\$ bin/server num_port filename.txt
- exécution client :
\$ bin/client server_name num_port

Veuillez bien mettre le même numéro de port pour le client et le serveur.

Limitation :

A défaut du temps, on n'a pas pu implémenter la fonction qui fait le tri d'une chaîne de caractères par ordre alphabétique concernant le protocole 1 et le protocole 2.

Conclusion :

Ce qui est intéressant dans ce projet est qu'il nous a permis de faire des réflexions pas que dans la programmation seulement, mais aussi dans la répartition des tâches pour finir dans les deadlines indiquées, le côté intégration, qui n'est pas aussi simple que l'on pensait. Et on peut des fois impacter même les décisions prises côté implémentation, on a pris conscience que la conception et une phase primordiale avant de se lancer dans le code.