Université Grenoble Alpes **Année 2022/2023**
L3 MIAGE

Projet Système/Réseaux

Décembre 2022

I. Objet du projet

Le but de ce projet est de réaliser en C une **application client/serveur TCP/IP** simplifiée.
Ici nous allons réaliser un serveur qui permet **l'interrogation d'un fonds documentaire**. Les informations sur les livres disponibles seront rassemblées sur ce **serveur**, que les **clients** interrogeront selon les besoins des utilisateurs.
Attention, il n'y a aucun aspect bases de données dans ce projet, le format texte qui sera utilisé pour le fonds documentaire est simpliste et fictif.

Représentation des informations disponibles.

Les livres disponibles seront **caractérisés par** : un *numéro* de référence dans le fonds documentaire, le *nom* de l'auteur (pour simplifier, nous supposerons 1 seul auteur), le *titre*, le *genre littéraire* (roman, nouvelle, poésie, récits,…), le *nombre de pages*, et une *appréciation* attribuée par les lecteurs précédents (A, B, ou C).
Ces informations seront rassemblées dans un *fichier de texte* sur le serveur, au format suivant : une ligne par livre, avec des champs représentant les caractéristiques listées ci-dessus, séparés par un "#" (voir exemple en Annexe, aussi disponible dans le moodle).

Interrogations possibles de la part des clients.

Les utilisateurs du service utiliseront votre application **client** afin d'émettre des requêtes auprès du serveur.
Attention à ne pas confondre ce que le serveur et le client s'échangent et ce que l'utilisateur saisit ou voit à l'affichage ! (le client peut formater des requêtes vers le serveur, et aussi formater les réponses du serveur pour un affichage convivial à l'utilisateur)

L'application devra offrir à l'utilisateur les possibilités suivantes :

* soumettre une *référence* et obtenir l'affichage des informations associées au *livre concerné* : nom de l'auteur, titre du livre, et genre. On affichera également le message "Moins de 300 pages" ou "Plus de 300 pages" selon le nombre de pages du livre.
  Mais **attention** aussi à ce que le client soit en mesure de délivrer à l'utilisateur un message clair dans le cas où la référence demandée n'existe pas (quel choix efficace ferez-vous pour ce qui sera renvoyé au client ?).
* soumettre un ou des *mot-clés*, et obtenir l'affichage des informations associées aux *livres dont le titre contient ce ou ces mots*, c'est-à-dire pour chacun d'eux : référence, nom de l'auteur, titre du livre, et genre. Si plusieurs réponses sont disponibles, elles seront présentées triées par ordre alphabétique des noms d'auteurs.
  Attention ici aussi à ce que le client soit en mesure de délivrer à l'utilisateur un message clair dans le cas où aucun livre ne safisfait la requête.
  Remarque importante : nous avons ici un cas où **plusieurs** données peuvent avoir à être transmises au serveur, et **plusieurs** résultats peuvent être fournis. Vous devrez **commencer** par décider du *protocole d'échange que le serveur et le client utiliseront pour ces transmissions multiples* : envoi en une seule fois (comment les données seront-elles distinguées ?), envoi en plusieurs fois (comment ?). Vous êtes libres de vos choix et ils **devront** être expliqués dans le rapport joint.
* soumettre un *nom d'auteur* et un *genre littéraire*, et obtenir l'affichage des informations associées aux *livres disponibles de cet auteur et pour ce genre* : référence et titre du livre. Attention à prendre en compte ici aussi toutes les remarques précédentes.
* soumettre un *nom d'auteur* et demander à l'application de *proposer un livre parmi les*

livres disponibles de cet auteur, sélectionné selon l'un des critères suivants (retenir par exemple le premier livre qui satisfait le critère, dans le cas où plusieurs le satisfont) :

- le nombre de pages minimum,
- la meilleure appréciation des lecteurs.

/!\ Pour ces interrogations, il faudra bien sûr réfléchir à mettre en œuvre au préalable toutes les **fonctionnalités de base** nécessaires, en favorisant un maximum de réutilisabilité.
Vous devez préparer cela, et l'intégration de ces fonctionnalités, **avant** de coder.
Les choix d'organisation logicielle ici **devront** être clairement expliqués dans le rapport joint.

II. Travail à réaliser.

Nous allons mettre en œuvre cette application **sur les PC Linux de l'UFR** (attention, les présentations seront faites exclusivement sur ces machines), en documentant correctement votre code (documentation des fonctions, et commentaires pertinents dans le code).
Procéder **selon les étapes suivantes** :

(Phase 1) Réaliser tout d'abord le **noyau de client/serveur TCP/IP** en suivant les indications qui vous ont été données au lancement de ce projet : en particulier, suivre les schémas donnés dans le petit mémento, et identifier les différentes phases dans votre code.
Le serveur doit être capable de communiquer avec plusieurs clients simultanément et, à ce stade, vous vérifierez juste que clients et serveur arrivent à s'échanger une information élémentaire (un entier, une chaîne de caractères,…).

Pour tester cette première version[1] vous pouvez commencer à tester avec serveur et clients sur la même machine, en utilisant l'interface de loopback (localhost), mais il faudra aussi tester réellement **avec serveur et plusieurs clients simultanés en réseau**.
Réaliser tous ces tests **avant** de passer à la phase suivante.

Cette première phase devra être **terminée**, et correctement documentée, dès lundi 28/11 au soir.

(Phase 2) **Concevoir** le **protocole d'échange** selon lequel serveur et client pourront interagir : les composantes d'une requête/réponse seront-elles transmises séparément ou via une seule requête/réponse composite ? comment le serveur contrôlera-t-il les échanges ? comment la requête/réponse sera-t-elle traitée à sa réception ?...
/!\ Toute cette réflexion devra figurer dans votre rapport final (voir plus loin).

Réaliser les primitives permettant la mise en œuvre de ce **protocole d'échange**. A ce stade, vous vérifierez que ces primitives fonctionnent comme attendu. Elles pourront être intégrées dans le noyau de client/serveur de la phase 1 afin de tester si un dialogue (données quelconques pour l'instant) peut s'établir correctement entre le serveur et le client.

(Phase 3) Concevoir (selon vos propres choix) et réaliser les **fonctionnalités de base** qui seront nécessaires pour les interrogations décrites ci-dessus.
Tester chacune de ces fonctions de base séparément et assurez-vous que les résultats sont justes.
/!\ La conception de ces fonctions devra également être décrite dans le rapport final (décrire les principes, la conception logicielle, pas le code).

NOTE. Si vous avez des difficultés, vous pourrez dans un premier temps ne prévoir que ce qui est nécessaire pour le premier type d'interrogation décrit ci-dessus (recherche d'un livre selon sa référence), puis passer à la phase 4 (et revenir compléter dès que la phase 4 sera au point).

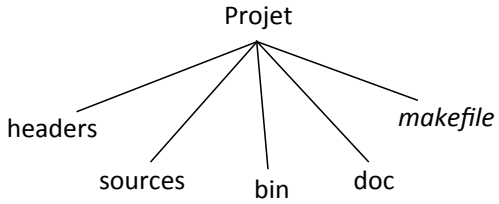
(Phase 4) Enfin, **intégrer le noyau client/serveur avec toutes ces fonctionnalités**.
/!\ Attention, il faudra avoir prévu cette intégration dès les choix faits en phases 2 et 3 !

Bien entendu, le **serveur** devra recevoir en paramètre à l'appel le *numéro de port* dédié et le *nom du fichier* contenant les informations sur le fonds documentaire, et le **client** devra recevoir en

[1] de même pour tester les versions suivantes

paramètre à l'appel le *nom (ou l'adresse IP) du serveur* et le *numéro de port* dédié.

(Phase 5) Organiser toute votre application selon une arborescence décrite ci-dessous :



Les fichiers .h seront placés dans le répertoire *headers*, et les fichiers .c dans le répertoire *sources*.
Le répertoire *bin* sera vide, et destiné à recevoir les exécutables produits par le makefile (écrire le makefile en conséquence).
Le makefile, placé à la racine du projet, permettra de compiler le serveur et le client.
Le sous-répertoire *doc* contiendra au minimum le pdf de votre rapport, et un ou des fichiers exemples permettant de tester votre application.

==========

Le projet est à réaliser par groupe de 3 ou 4 personnes (formés lors de la séance de démarrage du 28 nov.).
Il donnera lieu à :

- une séance de démonstration + questions pour chaque groupe (le **2 décembre**), présence de tous les membres du groupe obligatoire,
- la remise de vos sources (soigneusement commentés) + un rapport en pdf.

L'état courant du **rapport** devra être remis (format papier) **au moment de la démonstration**.

La remise finale des sources + rapport en pdf pourra se faire **jusqu'à 23h30 le 2/12**.
Les projets non remis à cette échéance précise ne seront pas corrigés (note = 0). Aucun envoi par mail ne sera pris en compte.

Comme pour le projet Système, vous remettrez un ***unique fichier groupeX.tar.gz*** où X est le numéro du groupe. Attention, il devra être produit à partir d'un répertoire lui-même appelé *groupeX*.
(le non respect de toutes les consignes de remise entraîne une pénalisation de la note)

==========

Le **rapport** sera conçu selon le plan de votre choix. Nous devrons **au moins** y trouver, présenté clairement :

- une introduction (importante !) présentant, avec vos propres mots, le **sujet** du projet et les **réalisations attendues**, ainsi que **ce que ce projet permet d'illustrer/approfondir** en ce qui concerne vos enseignements système et réseau,
- une présentation de la **conception logicielle** globale,
- une présentation du **protocole d'échange** client/serveur défini et des aspects techniques de sa mise en œuvre dans votre application,
- une présentation de la conception et de la mise en œuvre des **fonctionnalités** servant aux interrogations par le client,
- des exemples **d'expérimentations** (avec résultats expliqués) - penser à préciser comment compiler et exécuter votre application,
- une **conclusion**. Entre autres, elle pourra aussi inclure un récapitulatif de ce qui a / n'a pas été réalisé, ou bien des extensions (si vous en avez faites).

La longueur du rapport sera de l'ordre d'une quinzaine à vingtaine de pages (caractères 11 ou 12 points, simple espacement, dessins de taille suffisante pour être lisibles - pas surdimensionnés !).

ANNEXE - Fichier exemple

```
10#Jack London#Aventures en mer#recits#1024#B
20#Moliere#Dom Juan#theatre#192#B
30#Stendhal#La chartreuse de Parme#roman#550#B
40#Victor Hugo#La legende des siecles#poesie#1030#B
50#Honore de Balzac#La peau de chagrin#roman#416#A
60#Jack London#L'appel de la foret#roman#158#A
70#Honore de Balzac#L'auberge rouge#nouvelle#80#B
71#Honore de Balzac#Le pere Goriot#roman#384#C
80#Stendhal#Le rouge et le noir#roman#640#A
90#Ernest Hemingway#Le vieil homme et la mer#roman#144#A
100#Victor Hugo#Les contemplations#poesie#608#A
110#Moliere#Les fourberies de Scapin#theatre#192#A
120#Victor Hugo#Les miserables#roman#352#A
130#Moliere#Les precieuses ridicules#theatre#128#B
140#Victor Hugo#Les travailleurs de la mer#roman#674#C
141#Victor Hugo#Notre-Dame de Paris#roman#730#A
142#Victor Hugo#Odes et ballades#poesie#800#B
143#Victor Hugo#Quatre-vingt-treize#roman#575#B
150#Honore de Balzac#Un drame au bord de la mer#nouvelle#45#A
```