

On demande de simuler la location de Livres.

Pour cela, vous lancez un processus **QtDossier2013**, et vous disposez de l'écran de saisie suivant :

Le nom de différentes variables est donné en fin d'énoncé.

Une fois identifié, le client rentre les livres empruntés. Ensuite, le client demande la location de livres soit par son numéro dans le stock (à condition qu'il le connaisse) soit en fonction de l'auteur et alors, il choisit dans une liste.

Une fois la commande passée, le bouton <Terminer> permet de clôturer votre location et de passer à un client suivant.

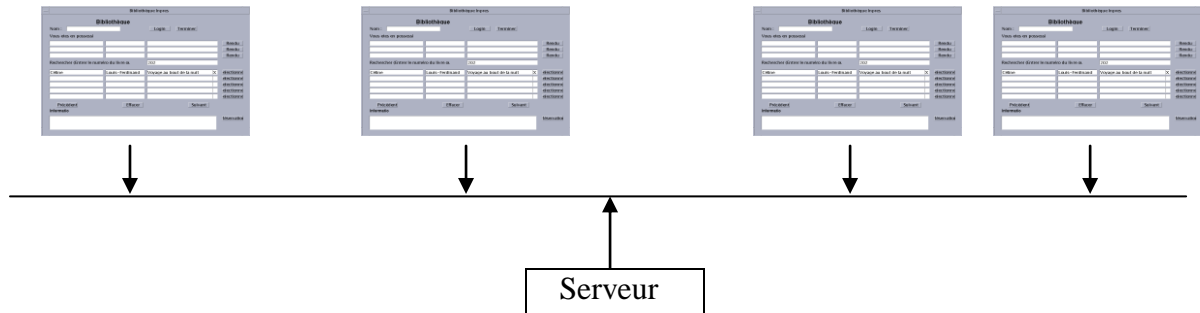
De plus, dans une zone Information, il peut recevoir des renseignements divers.

Par exemple:

- La bibliothèque va fermer et vous devez partir.
- Le livre sélectionné n'est plus en stock, faut-il le réserver ?
- Le livre est déjà réservé et il n'est pas possible le réserver encore.
- Le livre que vous avez réservé vient de rentrer.
- Une rencontre avec un auteur est organisée ce samedi à 13H.
- ...

Pour cela, le processus QtDossier2013 communique avec un processus Serveur, et c'est celui-ci qui s'occupe de tout.

Le processus QtDossier2013 communique avec le processus Serveur par *file de messages*.



Dans un premier temps :

Le processus Serveur :

Le processus Serveur crée les ressources nécessaires (pour le moment, il n'y a qu'une file de messages), ouvre le fichier StockLivreSUN.dat .

Il travaille suivant la technique des processus **démon**.

C'est-à-dire, il attend une requête et la transmet aussitôt à un processus indépendant qui la traite.

A la réception d'un message de QtDossier2013, il analyse la requête et effectue le traitement correspondant.

IDENTIFICATION :

cas d'un nouveau client, il transmet la liste des livres empruntés (si il y en a).

Le client ne peut être connecté qu'une seule fois.

RECHERCHE :

cas d'une recherche, il lance un processus AccesLivre qui effectue la recherche du (des) livre(s). En fonction de son numéro ou de son auteur. (Attention un auteur peut avoir plusieurs livres dans la bibliothèque)

SELECTION :

cas d'une réservation de nouveaux livres, il la transmet au processus AccesLivre qui effectue la location dans le fichier (qui contient un champ Libre pour chaque livre)

RETOUR : (bouton Rendu)

Idem, le champ Libre sera positionné.

FINCLIENT :

cas où le client termine sa commande.

CONFIRME : (bouton Reservation)

Cas ce cas, le livre est réservé dans le cas où il ne l'est pas déjà.

Pour cela, on dispose des fichiers et processus suivant :

Le fichier LitLivre.cpp :

Il ne sert à rien dans l'application, mais il permet de lire le contenu du fichier LivreSun.dat.

Note, un entier est encodé de manière différente sur un sparc et sur un pc. Ce qui explique pourquoi il y a 2 fichiers de données.

Le fichier Fichier.ini :

Dans lequel on définit les ressources nécessaires à l'application et dont les processus ont besoin.

A savoir : la clé des I.P.C., la définition des files de messages, les différentes requêtes.

Les fichiers Ecran.cpp et Ecran.h :

Ce fichier contient la définition des fonctions Trace() et TraceErreur() utilisées dans tous les processus.

On peut y définir d'autres fonctions tout aussi utiles.

Le processus QtDossier2013 :

Processus sans commentaire :

```
g++ -I /usr/local/qt/include -I /usr/local/qt/mkspec/default -o QtDossier2013  
Main.cpp FenetreSaisie.o moc_FenetreSaisie.o Ecran.o -lqt
```

Remarque : Ce processus est écrit en Qt, surtout ne pas modifier le .h. le fichier moc_FenetreSaisie().cpp est là pour indiquer les actions. Ne pas le modifier. Seules les fonctions du fichier FenetreSaisie() peuvent être modifiées.

Le processus Serveur :

Processus parlé plus haut qui gère les commandes.

C'est lui qui gère les ressources de l'application (les crée et les détruit) et qui lance les processus AccesFichier.

```
g++ -o Serveur Serveur.cpp Ecran.o
```

Le processus AccesFichier :

```
g++ -o AccesFichier AccesFichier.cpp Ecran.o
```

Ce processus est lancé par le Serveur lors de la première recherche et se termine lorsque le client se termine.

Il y a donc autant de processus **AccesFichier** que de clients ayant effectué une requête.

Ce qui implique une relation entre le pid du processus QtDossier2013 et celui du processus AccesFichier.

C'est le processus Serveur qui stocke dans une table cette correspondance.

Pour cela, lorsque le QtDossier2013 envoie un message avec la requête **IDENTIFICATION**, le serveur ajoute une entrée dans une table.

Lorsque QtDossier2013 émet une requête **RECHERCHE**, le processus Serveur lance le processus AccesFichier et associe son pid au processus QtDossier2013. (On a donc une correspondance entre les 2 processus pour la suite des évènements (**COMMANDE** (bouton Selection) ou **RESERVATION**))

Pour terminer, le processus QtDossier2013 termine sa commande en émettant la requête **FINCLIENT** qui permettra au processus Serveur de tuer AccesFichier correspondant et de le supprimer de la table.

Ce processus s'occupe de l'entièreté de la commande.

Il effectue donc la recherche dans le fichier StockLivreSun.dat, mais aussi il positionne le champ Libre de la structure lorsqu'un livre est réservé.

Le processus AccesFichier se termine avec le client.

Remarque : Lorsque le processus AccesFichier se termine, il doit être supprimé de la table des processus.

Vous disposez des fichiers StockLivreSUN.dat (StockLivrePc.dat)

La suite, maintenant...

Première étape, et la plus simple.

Il faut s'identifier. Un utilisateur de la bibliothèque doit être connu.

Pour cela, le Serveur recherche si l'utilisateur est inscrit dans un fichier Membre.dat.

(un fichier texte suffit, une ligne par utilisateur. On peut le modifier par un éditeur quelconque.)

Un message sera transmis si c'est un membre ou non.

Deuxième étape.

Construire la table effectuant la relation entre l'utilisateur et le processus AccesFichier lancé.

NomUtilisateur	PidQtDossier2013	PidAccesFichier
MerceD	1234	2345
Wagner	4321	0

Lors du **Login**, il faut vérifier que l'utilisateur n'est pas déjà dans la table. En effet, il ne peut pas louer des livres à 2 endroits différents en même temps.

Dans ce cas, MerceD a déjà effectué une recherche, Wagner est identifié mais n'a pas effectué de recherche.

Pour la suite, le processus AccesFichier est créer, il suffit donc de lui transmettre les requêtes.

Lors de **Terminer**, il doit être supprimé de la table après avoir tué le processus AccesFichier correspondant. (Dans ce cas, il devient zombie, et il faut le supprimer de la table des processus)

Ensuite,

Il faut effectuer la réservation proprement dite. (cas **Sélectionner**). Dans ce cas, le livre doit être libre, le processus AccesFichier doit donc écrire la modification (champ Libre dans la structure LIVRE) dans le fichier.

Si le livre est déjà loué, il peut éventuellement le réserver.

Dans ce cas, le numéro du livre ainsi que le nom de l'utilisateur doit être écrit dans un fichier Reservation.dat.

Pour des raisons propres à la bibliothèque, un livre ne peut être réservé 2 fois.

Dans les 2 cas, un message sera envoyé à l'utilisateur.

Dans le cas d'une location, celle-ci sera écrite dans un fichier Location.dat dans le but évident de connaître qui est en possession du livre et depuis quand).

Le fichier contiendra donc les renseignements suivant

Date	NomUtilisateur	NumeroLivre	Rentre
3	Bidon	0	0
3	Merce	143	False
3	Merce	202	False
3	Wagner	21	False

Pour simplifier, la date sera représentée par un entier (incrémenté de 1 chaque jour)

La première ligne contient un enregistrement bidon, dont le premier champ sera la date (c.à.d. le numéro du jour)

(Note, pour des raisons de tests, une journée de dure pas 24h. Elle sera déterminée par la suite)

Étape :

Maintenant qu'une location est effectuée, il faut rapporter les livres quand ceux-ci sont lus.

Lors du **login** suivant, il faut parcourir le fichier Location.dat afin de connaître la liste des livres loués par l'utilisateur et de lui permettre de les rendre.

(Vous avez compris qu'un utilisateur ne peut être en possession que de 3 livres).

Lorsqu'un livre est rentré, il disparaît de la liste des livres en possession de l'utilisateur, (ne pas oublier de le signaler dans le fichier Location.dat et dans le fichier Livre.dat)

LA SUITE, DEMAIN

Le nom des variables sera pour lundi, je n'ai pas le source avec

lineSaisieNom
lineRecherche
textMessage

lineNom[3]
linePrenom[3]
lineTitre[3]

lineNomOk[5]
linePrenomOk[5]
lineTitreOk[5]
lineReserveOk[5]

ButtonLogin
ButtonReservation
ButtonEffacer
ButtonTerminer
ButtonRendu[3]
ButtonSelectionnerOk[5]

ButtonPrecedent
ButtonSuivant
ButtonEffacer