PR 3602
– Approfondissement en Informatique –
Système d’exploitation

ADAM Baptiste
GÉLIN Antoine

Rendu le mardi 30 avril 2019

Table des matières

Compréhension du projet

Afin de commencer ce projet correctement, nous nous sommes posé des questions afin de s'assurer que nous comprenions correctement ce que nous allons faire.

1/ Combien de processus seront créés dans le projet ?

Cela dépend du nombre de banques dans le projet ainsi que le nombre de terminaux par banque.

Il y aura un seul serveur interbancaire qui sera dans un processus propre pour tout le projet. Ensuite pour chaque banque on aura un serveur d'acquisition (1 processus par banque) et un serveur d'autorisation (1 processus par banque aussi).

À chaque banque sera relié un nombre non défini de terminaux (n par banque).

Ainsi le nombre total de processus si on a m banque avec chacun n terminaux sera : $Nb_{banque} = 2m + mn + 1$.

2/ Combien d'exécutables allez-vous programmer ? Est-ce que un seul exécutable est acceptable ?

Nous allons développer 4 exécutables:

- le serveur d'acquisition
- le serveur d'autorisation
- le terminal de paiement
- le serveur interbancaire

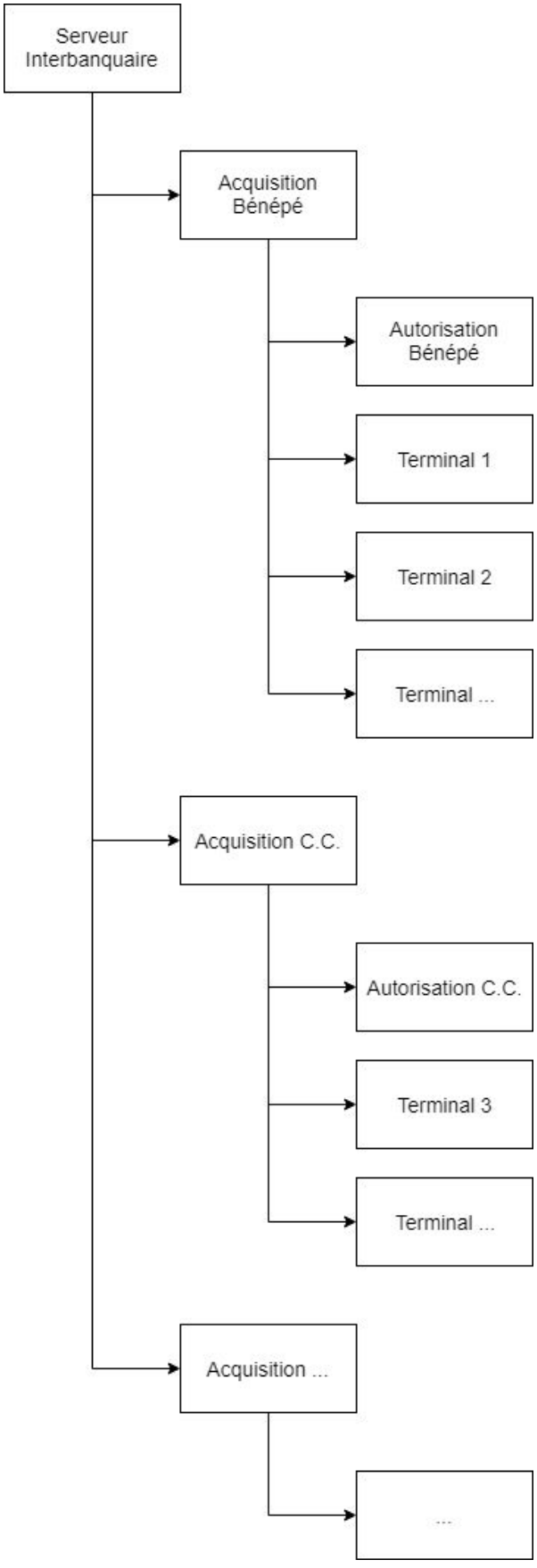
3/ Qui (quel processus) est créé par qui ?

On part du serveur interbancaire car c'est le centre du réseau et doit connaître toutes les banques participantes au réseau.

Ensuite le serveur interbancaire va créer le serveur d'acquisition pour le nombre de banques participantes.

Chaque serveur d'acquisition représente une banque qui se chargera de créer le serveur d'autorisation ainsi que les terminaux de chaque banque.

4/ Dessinez l'arbre des processus du projet.



5/ Combien de tubes sont nécessaires pour le projet ?

Pour chaque communication entre deux processus, les informations doivent aller dans les deux sens. Il sera donc nécessaire d'avoir à chaque fois 2 tubes pour chaque communication.
On aura donc :

- une communication entre le serveur interbancaire et le serveur d'acquisition de chaque banque (2 sens)
- une communication entre le serveur d'acquisition et le serveur d'autorisation à l'intérieur de chaque banque (2 sens)
- une communication entre le serveur d'acquisition et chaque terminal leur appartenant au sein de chaque banque (2 sens aussi)

En tout on aura $Nb_{tubes} = 2(m + mn + 1)$ avec m le nombre de banques et n le nombre de terminaux dans chaque banque.

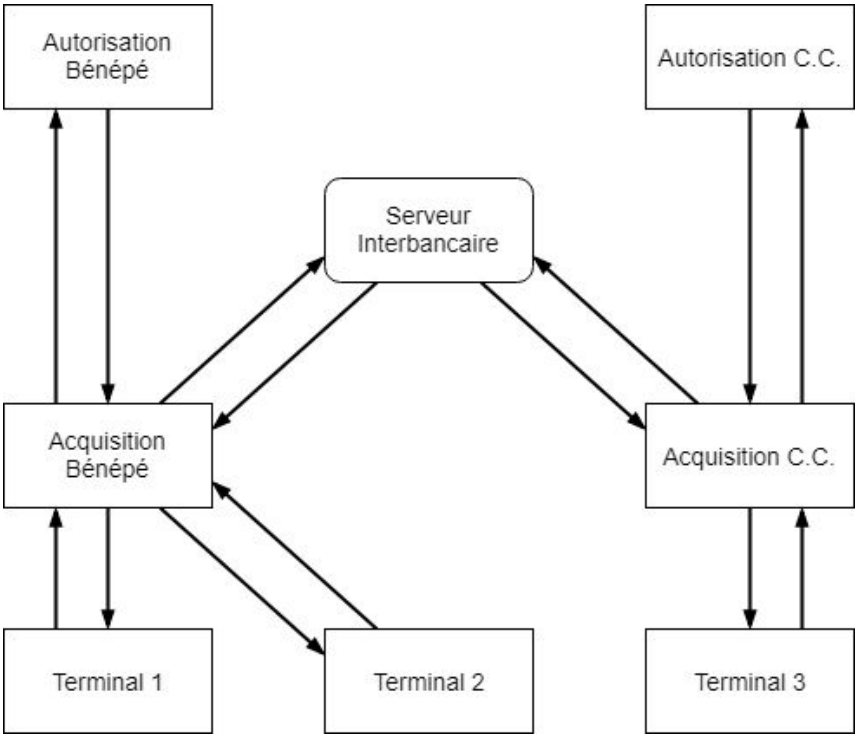
6/ Quels types de tubes allez-vous utiliser ?

Des tubes non nommés car n'avons pas besoin de stocker des grandes données, et surtout pour un souci de confidentialité, nous ne voulons pas que des processus externes aient accès à nos tubes.

7/ Qui (quel processus) crée quels tubes ?

"Serveur Interbancaire" commence par créer les tubes le liant avec les processus "Acquisition". Ensuite chaque processus "Acquisition" crée les tubes le liant aux processus "Autorisation" et aux processus "Terminal".

8/ Dessinez vos processus et vos tubes.



Décomposition du projet

Basé sur notre compréhension, nous avons décidé de segmenter notre projet en plusieurs versions à la complexité croissante. Les raisons de ces segmentations seront expliquées plus tard, dans leurs parties correspondantes.

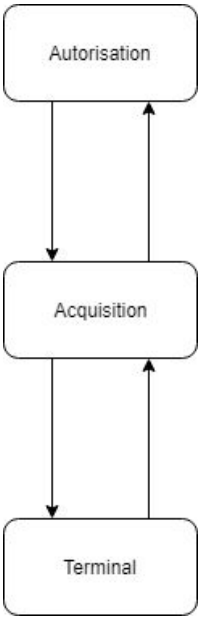
La première version, la version basique, comporte un Serveur d'Acquisition, un Serveur d'Autorisation et un Terminal. Lors de la seconde version, nous mettons plusieurs Terminaux en relation avec le Serveur d'Acquisition. Lors de la troisième version, la structure reste la même par rapport à la version antérieure, mais la notion de parallélisme est introduite. Enfin, lors de la quatrième version, nous ajoutons le Serveur Interbancaire et nous le mettons en relation avec plusieurs Serveurs d'Acquisition dont la structure est basée sur la version précédente.

I/ Version basique

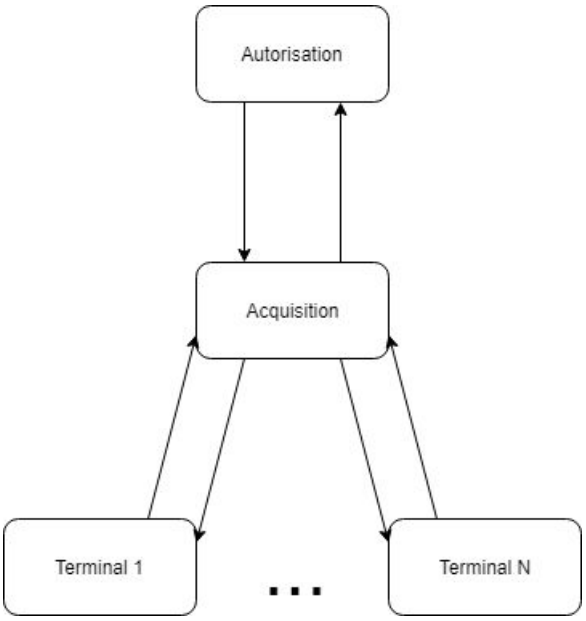
Cette version la plus simplifiée possible possède :

- 1 Serveur d'Autorisation
- 1 Serveur d'Acquisition
- 1 Terminal

Le but principal de cette version est de vérifier que les connexions des tubes se fait correctement pendant que le code est encore simple.



II/ Ajout de plusieurs terminaux



Cette fois ci, le Serveur d'Acquisition traite N Terminaux à la fois. Le but est de réussir à créer tous les Terminaux et de les relier correctement au Serveur d'Acquisition.

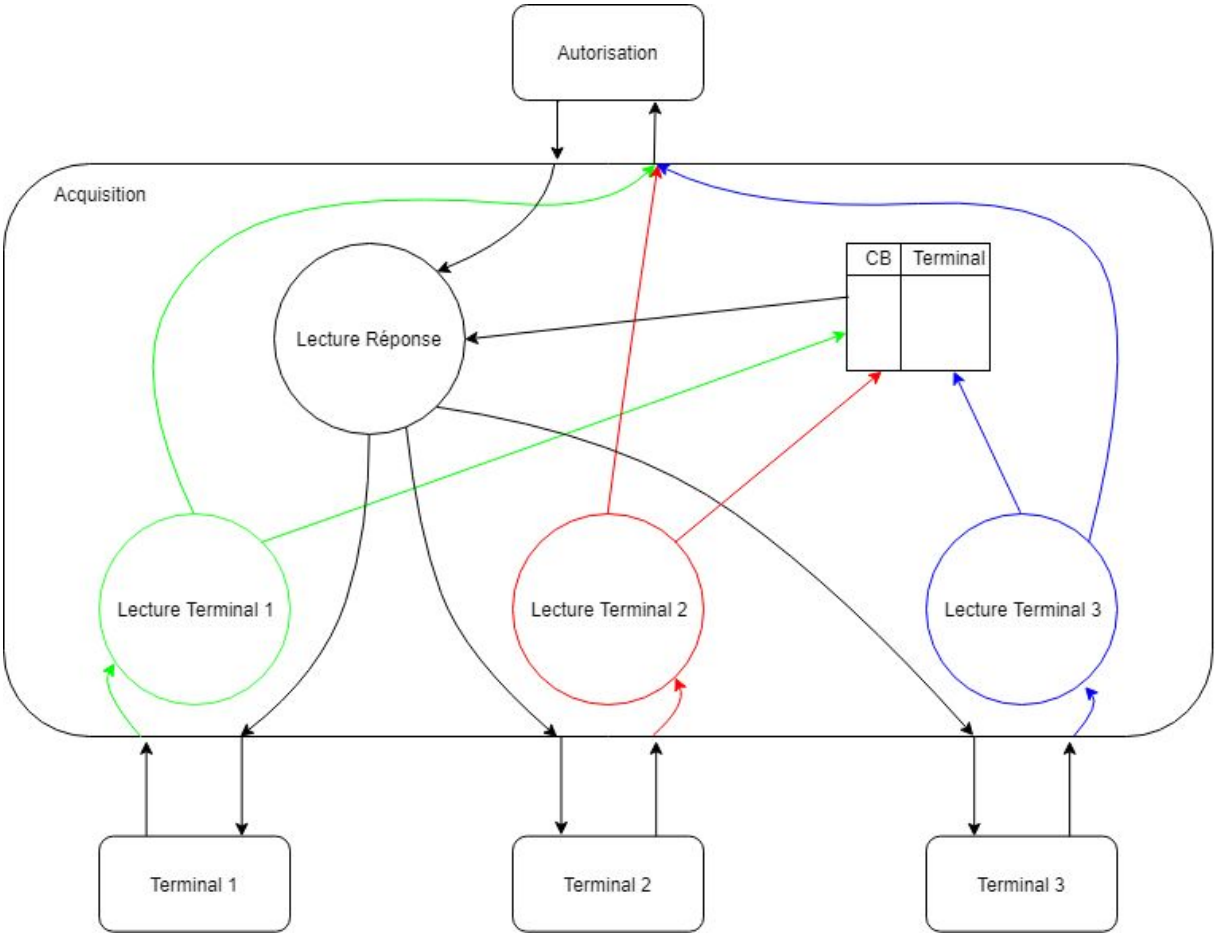
De cette façon, la structure se rapproche de ce que nous voulons : une banque possède plusieurs Terminaux. Le Serveur d'Acquisition traite successivement chaque Terminaux dans une boucle for.

III/ Ajout du parallélisme

Parce que les Terminaux sont traités successivement dans une boucle for, une notion d'ordre est induite : le Terminal 1 doit être traité avant le Terminal 2. Cela engendre un problème qui peut être décrit comme suit :

Après que les terminaux aient fonctionnés pendant un certain temps, le Terminal 1 n'envoie plus de demande (parce qu'il n'y a plus de client qui l'utilise) mais le Terminal 2 est toujours en fonction et attends une réponse à sa plus récente demande. Mais, comme le Serveur d'Acquisition traite les Terminaux dans l'ordre, il va bloquer sur le terminal 1 et attendre que celui-ci lui envoie une demande pour la traiter avant de passer au terminal 2 (qui attend déjà depuis un certain temps).

Pour ce faire, il faut ajouter une fonctionnalité de parallélisme dans le fonctionnement du Serveur d'Acquisition pour enlever cette notion d'ordre. Voici le schéma sur lequel nous nous sommes basé :



Pour N Terminaux, il y a N+1 threads créés :

- **Lecture Terminal [i]** : N threads servent à lire les demandes envoyés par le Terminal auquel ils sont affectés.
- **Lecture Réponse** : 1 thread sert à lire les réponses envoyé par le Serveur d'Autorisation et à le rediriger vers le bon Terminal.

A chaque fois qu'un **Lecture Terminal** reçoit une demande de son Terminal, il stocke dans un tableau le numéro de carte de la demande ainsi que le *File Descriptor* d'écriture de son Terminal. Cela permet à **Lecture Réponse** de savoir à quel Terminal envoyer la réponse qu'il vient de recevoir du Serveur d'Autorisation en se basant sur le numéro de carte pour trouver le bon *File Descriptor.*

Pour stocker les données partagées au sein d'acquisition, nous avons construit une structure nous permettant d'associer rapidement un numéro de CB au FileDescriptor d'un Terminal. Cela nous permet d'aller plus vite dans nos développements.

Nous avons deux sections critiques: une lors de la copie des arguments lors de la création des threads d'acquisition et une pour s'assurer de la constance des informations du tableau partagé.

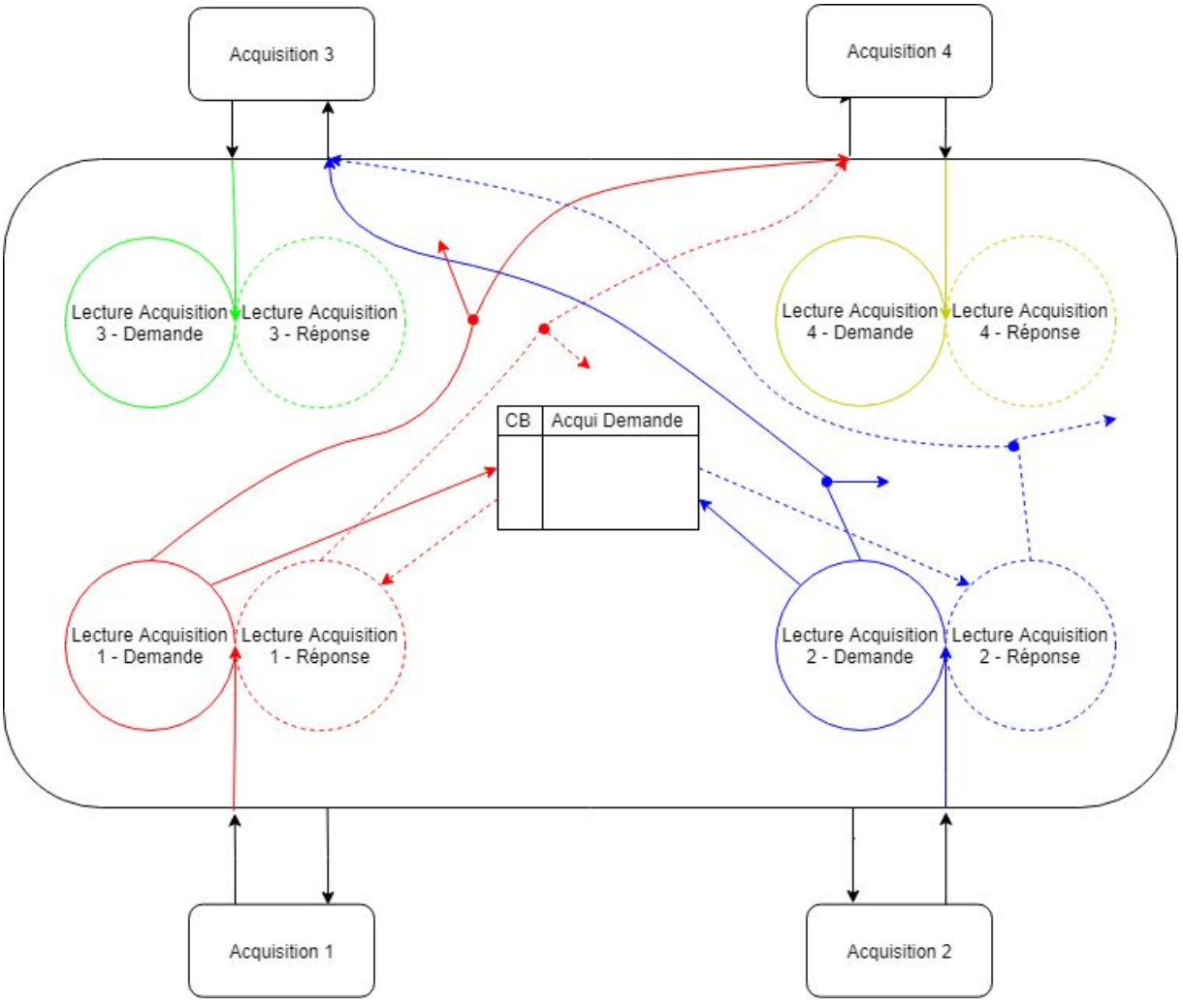
IV/ Création du réseau interbancaire

Le Serveur Interbancaire répond aux mêmes problématiques que le Serveur d'Acquisition, il possède donc une structure très similaire. La différence principale est qu'il ne discute pas entre N terminaux et un Serveur d'Autorisation, mais entre M Serveur d'Acquisition.

En effet, lorsque le Serveur d'Acquisition reçoit une demande d'une carte qui n'est pas dans sa banque, il doit rediriger ladite demande vers le Serveur d'Acquisition correspondant, cela se fait au travers du Serveur Interbancaire.

Le Serveur Interbancaire et les Serveurs d'Acquisition doivent donc pouvoir accéder à une liste de correspondance entre les Cartes bancaires et les banques.

Voici le schéma sur lequel nous nous sommes basé :



Pour des raisons de lisibilités, toutes les flèches n'ont pas été matérialisées

Lorsqu'un thread reçoit un message du Serveur d'Acquisition auquel il est affecté il va vouloir envoyer et rediriger le message vers le serveur d'autorisation de la banque concernée par la demande. Ce thread du serveur interbancaire saura qu'il s'agit d'une demande grâce à l'intitulé du message (Demande)

Dans un premier temps, il détermine vers quel serveur envoyer la demande grâce à la liste de correspondance carte-banque et stocke dans un tableau le *File Descriptor* d'écriture de son Serveur d'Acquisition duquel est rattachée la demande du terminal (banque externe). La demande est elle envoyée au serveur d'autorisation de la banque emettrice de la carte.

Dans un second temps, il reçoit une réponse depuis un serveur d'acquisition et doit ensuite obtenir dans le tableau le *File Descriptor* du Serveur d'où origine cette demande comme vu précédemment et y envoyer la réponse vers le serveur d'acquisition original qui la dirigera naturellement vers le terminal correspondant à l'aide des méthodes développées antérieurement.

On peut s'assurer que les demandes du tableaux ne sont pas réécrites par dessus par de nouvelles demandes en utilisant des sémaphores.

Ajouts supplémentaires

- Nous avons utilisé la commande xterm afin de lancer chaque Terminaux et Serveur dans une fenêtre séparée pour clairement voir qui écrit quoi.
- Ajout d'une interface pour lancer le programme (programme main.c) qui demande les paramètres voulus dans une interface homme machine. Elle génère aussi les registres de cartes correspondants aux nombres de banques.
- Au cours de nos développements nous avons appris à utiliser les outils GDB et Valgrind. Bien qu'annexe au programme développé, ils nous ont aidés à de nombreuses reprises à nous rendre compte de ce qu'il se passait vraiment.

Problèmes rencontrés

Lors de l'implémentation du serveur interbancaire, un dysfonctionnement dans l'initialisation des terminaux les font partager leurs File Descriptors entre banques. Après avoir réfléchi au problème nous n'avons pu le résoudre.

Le programme ne marche donc qu'en indiquant une seule banque (signalé dans main.c)