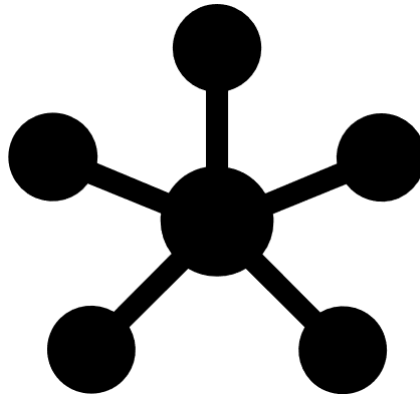




DEPARTEMENT INFORMATIQUE
DE LA FACULTE DES SCIENCES

Guilhem Blanchard (20134661), Yanis Allouch (21708237)

Rapport du Projet : Système de réservation distribué



**HMIN105M — Principes de la programmation
concurrente et répartie**

Référent: Hinde Bouziane

2020

Table des matières

1	Cahier des charges	3
----------	---------------------------	----------

Introduction

La présente introduction est tiré du sujet donnée.
Consigne importante pour le développement :

« Ce projet est à réaliser en binôme. Lire attentivement l'énoncé avant de commencer le travail. Pour la mise en place des connexions et communications distantes, vous utiliserez le protocole TCP/IP et les fonctions C utilisées en cours. Le plagiat est strictement interdit. »

Nous voulons mettre en place un système de réservation distribué développé en C sous un système UNIX.

« L'idée de ce projet s'inspire d'un système de réservation de ressources de calcul et/ou de stockage sur une plate-forme de grille ou de cloud. »

« Il a pour objectif de permettre à des clients de louer des puissances de calcul ou des espaces de stockage distants répondant à des besoins spécifiques (exemple : exécuter une simulation scientifique sur une architecture distribuée de processeurs et de mémoire pour stocker (le temps de la location) les données traitées et produites). »

« Dans le système à mettre en oeuvre, un client aura la possibilité de louer des ressources, soit en mode exclusif (les ressources louées sont utilisées par un seul client pendant toute la durée de la réservation), soit en mode partagé (les ressources louées peuvent être utilisées en même temps par plusieurs clients). »

1 Cahier des charges

Voici un aperçu général des tâches qui ont été repérées dans le sujet :

→ **Le rôle du serveur**

- mettre en place un espace partagé représentant l'ensemble des ressources disponibles et louées.
- gérer les accès clients à cet espace.
- maintenir cet espace dans un état cohérent.
- diffuser toute modification à tous les clients.

→ **Le rôle des clients**

- est d'échanger avec le serveur pour effectuer des réservations et libérations de ressources.
- avoir en continu une vue de l'état global de toutes les ressources offertes par le système de réservation.

Nous pouvons ainsi préciser les **caractéristiques techniques** importantes du serveur et du client :

- **Serveur** concurrent (mixte) : fork + threads
 - **Le parent** (thread principal)
 - * responsable de la mise en place de l'état initial du système de réservation.
 - * attendre les connections des clients.
 - **Un fils**, (thread fils) met en oeuvre des **threads** pour paralléliser le traitement de son client.
 - * maintient l'état des ressources cohérent.
 - * diffuse l'état des ressources.
 - * traite et communique exclusivement avec 1 client. Il attend une demande de
 - réservation de ressources.
 - libération de ressources.
 - * envoie à son client, toutes les modifications issues des autres clients.
- **Client**, met en oeuvre des **threads** pour paralléliser pour
 - recevoir et visualiser l'état du système de réservation dès sa connexion au serveur.
 - en boucle, saisi au clavier, envoyer des demandes de
 - * réservation de ressources.
 - * libération de ressources.

Présentons les caractéristiques des ressources mise à disposition d'une réservation :

- le système possède plusieurs site, identifié par un *Label* (unique).
- un *nombre entier* de processeurs.
- un *nombre entier* de capacité de stockage.

On peut donc proposer dans un premier temps, la structure C suivante qui

modélise nos ressources :

```
1 struct ressources {  
2     char[255] label;  
3     int nbProcesseurs;  
4     int nbCapaciteStockage;  
5 };
```

Ensuite, nous apprenons dans l'énoncé qu'un client peut :

- réserver de manière exclusive ou non une ressource.
- plusieurs ressources différentes à la fois.

Sachant que l'état du système est géré par un serveur concurrent léger, on peut donc utiliser les composantes suivantes , pour sauvegarder tous les échanges entres clients et serveur :

1. un tableau de toutes les ressources disponibles, informant de leur maximum.
2. un tableau des ressources **exclusifs** utilisées par, informant sur la quantité de ressources utilisé par site en mode exclusifs et de l'utilisateur ayant initié la réservation.
3. un tableau des ressources **partagées** utilisées, informant sur la quantité de ressources utilisé par site en mode partagé et de l'utilisateur ayant initié la réservation.

Il faut bien comprendre que nous faisons le choix que les champs initiaux respectifs du tableau 2 et 3 est initialisé à 0. De ce fait, un client réservant une ressource exclusif, vient augmenter la charge d'utilisation et par conséquent réduire la charge disponible dans le tableau de ressource exclusif si et seulement si la quantité totale moins la quantité exclusif utilisé, moins le maximum des ressources partagées est supérieur à la quantité demandé, toute ressource confondue.