

# Rapport sur les choix d'implantations du Serveur HTTP

## Tables des matières

Description.....	1
Traitement de la requête.....	2
Traitement de la réponse.....	2
Libération des ressources.....	2

## Description

En mode connecté (TCP), le schéma du serveur est le suivant :

Le serveur se met en écoute sur l'adresse et le port avec une liste de connexion où il y a une taille définie. Ensuite, il traite les demandes de connexion et alloue une ressource pour chaque client accepté. La ressource est un thread détaché afin qu'il s'exécute indépendamment.

Afin de traiter les clients un par un, j'ai décidé d'utiliser un verrou (exclusion mutuelle = mutex) pour thread (pthread\_mutex\_t).

## Traitement de la requête

Chaque requête étant formellement identique, j'ai décidé de créer un nouveau type **http\_request** qui est une structure regroupant les caractéristiques d'une requête. Cela me permet de manipuler les valeurs des en tête plus aisément.

La méthode, l'url, la version du protocole, et les en têtes noms et valeurs sont décortiqués afin de voir s'ils correspondent aux exigences du serveur HTTP.

Ainsi, si une erreur se produit lors de l'envoi de la réponse, le code de statut correspondra à l'erreur rencontrée lors du traitement de la requête.

## Traitement de la réponse

La réponse est aussi formellement identique, alors j'ai crée aussi un type **http\_response** avec les caractéristiques d'une réponse http. Les en têtes présent pour la plupart sont la date, le serveur qui envoie la réponse, le type de contenu, la taille du contenu, la date de dernière modification du contenu. Si le contenu n'a pas été modifié, alors le navigateur pioche dans son cache. Dans un premier temps, les en-têtes sont envoyés. Puis dans un second temps, c'est le fichier demandé qui est envoyé au navigateur client.

## Libération des ressources

Lorsque le serveur a terminé de traiter la requête du client, il ferme alors le descripteur de la socket de service du client.

De même que si le serveur reçoit un signal d'interruption, il ferme la socket de service du client ainsi que sa socket d'écoute sur l'adresse et le port.