

# **Rapport du projet aise**

**DRISS RYMA**

**M1 CHPS**

## Introduction :

Redis (Remote Dictionary Server) est une base de données en mémoire, open source. Conçu pour être rapide et efficace, Redis est souvent appelé un "store de données clé-valeur" ou un "magasin de structures de données".

## Objectif du projet :

Ce projet consiste à concevoir en langage C et mettre en œuvre un système de stockage clef-valeur performant en adoptant le protocole de communication de Redis.

- Implémenter des commandes de Base, ainsi d'autres commandes supplémentaires :  
Dans ce projet j'ai implémenter les commandes suivantes :
  1. PING : le client envoi PING et la réponse sera PONG.
  2. Commande SET : prend en charge l'ajout d'une clé valeur dans le contexte d'un serveur de stockage (SET clé val), la paire clé valeur sera ajouté au fichier cle\_valeur.data.
    - ➔ La fonction parcourt un tableau entrees pour rechercher si la clé est trouvée elle la met à jour sinon elle ajoute une nouvelle clé valeur.
  3. Commande GET : elle récupère la valeur associée à une clé.
    - ➔ La fonction recherche dans le tableau si une clé existe elle la renvoie sinon elle affiche clé non trouvée.
  4. Commande DEL : parcourt le tableau si la clé existe elle la supprime et supprime sa valeur.
  5. Commande DECR : a pour but de décrémenter la valeur associée à une clé.
    - ➔ La fonction utilise la variable current\_value pour stocker la valeur actuelle de la clé, elle recherche la clé dans le tableau si la clé est trouvée elle décrémente, sinon elle va décrémente à partir de zéro, elle affiche -1.
  6. Commande INCR : a pour but d'incrémenter la valeur associée à une clé.
    - ➔ La fonction utilise la variable current\_value pour stocker la valeur actuelle de la clé, elle recherche la clé dans le tableau si la clé est trouvée elle l'incrémente, sinon elle va incrémenter à partir de zéro, elle affiche 1.
  7. Commande EXISTS : elle parcourt le tableau, si la clé existe elle renvoi OK, sinon « no ».
  8. Commande ECHO : afficher un message passer à ECHO.
  9. Commande APPEND : elle ajoute du texte (valeur) a une clé, pour faire une concaténation.
  10. Commande RPush : ajoute des valeurs a une liste.
  11. Commande LRange : elle affiche les valeurs de la liste.
  12. Commande RENAME : elle renomme une clé, remplacer son nom par un autre.
  13. Commande QUIT : permet de quitter le client d'une manière propre (exit).
- Mettre en place un serveur qui écoute plusieurs clients (multi clients) capable de gérer les requêtes client :
  - ➔ Pour gérer simultanément plusieurs connexions clients, j'ai utilisé des threads qui ont permis au serveur de traiter plusieurs requêtes de clients en parallèle.

➔ L'utilisation de threads nécessite souvent l'utilisation de mutex (verrous de synchronisation) pour protéger les sections critiques du code partagé entre les threads. Dans mon code j'ai utilisé un mutex pour protéger l'accès concurrentiel aux données partagées.

- Système de stockage performant : pour assurer le stockage j'ai utilisé un fichier `cle_valeur.data` afin de stocker les clefs et les valeurs.