

# rapport

January 14, 2024

## 1 Rapport

### 1.1 Macros et Fonctions Utilitaires

Le programme utilise des macros et des fonctions utilitaires pour simplifier le code et gérer les erreurs. Par exemple, la macro `CHK` vérifie le résultat d'une opération et génère une erreur si nécessaire. La fonction `raler` est utilisée pour afficher des messages d'erreur détaillés.

```
#define CHK(op)      do { if ((op) == -1) raler (1, #op) ; } while (0)
#define CHKN(op)    do { if ((op) == NULL) raler (1, #op) ; } while (0)
```

```
\noreturn void raler (int syserr, const char *fmt, ...)
```

La structure `ServeurInfo` est utilisée pour stocker les informations relatives à un serveur. Elle comprend les champs suivants : `* name` : Nom du serveur. `* address` : Adresse IP du serveur. `* port` : Numéro de port du serveur.

### 1.2 Fonctions Principales du code Image

- `void usage (char *argv0)` La fonction `usage` est appelée en cas d'utilisation incorrecte du programme, affichant un message d'erreur expliquant le format d'utilisation correct.
- `void WriteInfoServeur(const char *serverName, const char *serv_adrIPv4, int serv_port)`  
Cette fonction enregistre les informations du serveur dans un fichier de configuration. Elle prend le nom du serveur, l'adresse IPv4 et le port en paramètres, crée ou ouvre le fichier de configuration, puis écrit les informations.
- `char *base (char *chemin)` La fonction `base` extrait le nom du fichier à partir d'un chemin complet.
- `void RepSendImageToServer(int clientSocket, char *requete, char *rep)` Cette fonction gère la réponse à une requête d'envoi d'image au serveur.
- `void RepListeImages(int clientSocket, char *rep)` Cette fonction gère la réponse à une requête de liste des images présentes sur le serveur.
- `void RepTesterExistenceImage(int clientSocket, char *requete, char *rep)`  
Cette fonction gère la réponse à une requête de test d'existence d'une image sur le serveur.
- `void RepSupImage(int clientSocket, char *requete, char *rep)` Cette fonction gère la réponse à une requête de suppression d'une image sur le serveur.

- `void RepRecupImage(int clientSocket, char *requete, char *rep)` Cette fonction gère la réponse à une requête de récupération d'une image depuis le serveur.
- `void serveur(int clientSocket, char *rep)` Cette fonction est appelée une fois que la connexion avec un client est établie, elle lit la requête du client et appelle la fonction appropriée pour traiter la demande.
- `void Connexion(const char *serv_adrIPv4, int serv_port, int socketType, char *rep)` Cette fonction configure le socket du serveur, le lie à une adresse et un port, puis entre en mode d'écoute. Elle accepte les connexions entrantes et appelle la fonction serveur pour traiter chaque connexion.

[ ]:

### 1.3 Fonctions Principales du code Tag

- `void usage (char *argv0)` La fonction `usage` est appelée en cas d'utilisation incorrecte du programme, affichant un message d'erreur expliquant le format d'utilisation correct.
- `void WriteInfoServeur(const char *serverName, const char *serv_adrIPv4, int serv_port)` Cette fonction enregistre les informations du serveur dans un fichier de configuration. Elle prend le nom du serveur, l'adresse IPv4 et le port en paramètres, crée le fichier de configuration, puis écrit les informations.
- `char *base (char *chemin)` La fonction `base` extrait le nom du fichier à partir d'un chemin complet.
- `void RepListeTagOfImage(int clientSocket, char *requete, char *rep, struct sockaddr_in clientAddr, socklen_t clientAddrLen)` Cette fonction gère la réponse à une requête de liste des tags associés à une image. Elle parcourt le répertoire `RepTag`, vérifie la présence de chaque tag associé à l'image, et envoie la liste des tags au client.
- `void serveur ( int clientSocket, char *rep, struct sockaddr_in clientAddr, socklen_t clientAddrLen )` La fonction serveur traite les requêtes du client. Elle analyse le type de requête et appelle la fonction appropriée pour y répondre.
- `void Connexion(const char *serv_adrIPv4, int serv_port, int socketType, char *rep)` La fonction Connexion crée une socket de serveur UDP, configure l'adresse, la lie à la socket, puis reçoit et traite une requête du client. Elle appelle la fonction serveur pour gérer la requête et ferme ensuite la socket.

### 1.4 Fonctions Principales du code Client

- `void usage (char *argv0)` La fonction `usage` est appelée en cas d'utilisation incorrecte du programme, affichant un message d'erreur expliquant le format d'utilisation correct.
- `int connectToServer(const char *address, int port, int socketType)` La fonction `connectToServer` permet d'établir une connexion avec le serveur Image en créant une socket TCP en fonction du type spécifié. Elle configure ensuite l'adresse du serveur et se connecte via TCP. En cas de succès, elle retourne le descripteur de fichier de la socket du client

- `void LireServeurInfo(const char *filename, struct ServeurInfo servers[])` La fonction `LireServeurInfo` lit les informations des serveurs à partir d'un fichier de configuration (`InfoServeur.cfg`), extrait les données nécessaires, et les stocke dans un tableau de structures `ServeurInfo`.
- `void ReqSendImageToServer(int clientSocket, unsigned char type, char *nom, char *reponse)` La fonction `ReqSendImageToServer` envoie une requête au serveur Image pour l'ajout d'une image. Elle prépare la requête en incluant le type, le nom du fichier, la taille du nom, la taille du contenu de l'image, et le contenu de l'image. Après l'envoi, elle reçoit la réponse du serveur et la traite.
- `void ReqListeImage(int clientSocket, unsigned char type, char *reponse)` La fonction `ReqListeImage` envoie une requête au serveur Image pour obtenir la liste des images. Elle récupère ensuite la réponse qui contient le nombre d'images et les noms des fichiers.
- `void ReqTesterExistenceImage(int clientSocket, unsigned char type, char *nom, char *reponse)` La fonction `ReqTesterExistenceImage` envoie une requête au serveur Image pour tester l'existence d'une image. Elle traite ensuite la réponse du serveur.
- `void ReqSupImage(int clientSocket, unsigned char type, char *nom, char *reponse)` La fonction `ReqSupImage` envoie une requête au serveur Image pour supprimer une image. Elle reçoit ensuite la réponse du serveur.
- `void ReqRecupImage(int clientSocket, unsigned char type, char *nom, char *reponse)` La fonction `ReqRecupImage` envoie une requête au serveur Image pour récupérer le contenu d'une image. Elle écrit ensuite le contenu de l'image sur la sortie standard.
- `void RepListeTagOfImage(int clientSocket, unsigned char type, char *nomImage, char *reponse, struct sockaddr_in *serv_addr)` La fonction `RepListeTagOfImage` envoie une requête au serveur Tag pour lister les tags associés à une image. Elle ne fonctionne pas.