Mini-Projet de
Programmation C avancée
Développement d'un système de messagerie

1. Introduction

L'objectif de ce projet est de mettre en œuvre les notions de programmation avancée en langage C vues en cours et en TP et de les consolider. Ce projet consiste à mettre en pratique les concepts de :

- fonctions
- structures de données
- pointeurs et allocation dynamique de la mémoire
- traitement de fichiers.
- listes chaînées
- appel de bibliothèques (API)
- réseaux

Votre travail consiste à mettre en place un programme C pour la gestion d'un **serveur de messagerie**.
Vous devez dans un premier temps présenter **un rapport résumant le projet dans un document soigné qui décrit :**

1. L'architecture de votre application
2. Les fonctionnalités de l'application.

En second lieu, vous serez amené à réaliser une **démo vidéo** (de 10 minutes) afin de présenter une démonstration pratique et fonctionnelle du projet.
La qualité de l'application sera jugée par sa capacité à tirer profit de **tous** les concepts vus en cours, par son **originalité**, par la **qualité du code** et par la **complexité des solutions proposées** (implémenter par exemple des listes chaînées au lieu de tableaux, des fonctions regroupées dans des fichiers séparés au lieu d'un seul, …).
Egalement, tout travail de recherche est récompensé.
Enfin, le projet peut être réalisé par binôme, **le plagiat sera sévèrement sanctionné**.

Il s'agit d'un travail évalué comme partie du « Contrôle Continu », à rendre sur l'adresse email suivante : projet.isitcom@gmail.com avant **30 avril 2024**.

2. Travail demandé

On vous demande de développer une application simulant un serveur de messagerie qui permet aux utilisateurs d'envoyer des messages les uns aux autres à travers leurs boîtes de messages.
Votre projet doit **au moins comporter ces deux parties** :

Partie Gestion des comptes des utilisateurs :

La gestion des utilisateurs dans le système de messagerie implique la création, la modification, la suppression des utilisateurs :

- ✓ Création de Comptes : Le système doit permettre l'ajout de nouveaux comptes utilisateurs. Lorsqu'un nouvel utilisateur souhaite rejoindre le système, il doit fournir un nom d'utilisateur et un mot de passe. Vous pouvez à ce niveau garantir qu'elles respectent les critères de sécurité requis.

- ✓ Authentification : Pour se connecter à leurs comptes, les utilisateurs doivent fournir leur nom d'utilisateur et leur mot de passe.

- ✓ Modification des Comptes : Les utilisateurs doivent avoir la possibilité de modifier leurs informations de compte, telles que leur nom d'utilisateur ou leur mot de passe. Cette fonctionnalité doit être sécurisée et nécessiter une ré-authentification avant d'effectuer toute modification.

- ✓ Suppression de Comptes : Les utilisateurs doivent avoir la possibilité de supprimer leur compte du système s'ils le désirent. Cette action doit être confirmée par l'utilisateur pour éviter toute suppression accidentelle de compte.

Partie Gestion des Messages :

La gestion des messages dans le système de messagerie intègre l'envoi, la réception, le stockage et la gestion des messages échangés entre les utilisateurs :

- ✓ Envoi de Messages : Une fois qu'un utilisateur a composé un message, il doit pouvoir le transmettre à un ou plusieurs destinataires. Vous pouvez ajouter ici des options pour la sécurité du message (cryptage, signature, …)

- ✓ Consultation de la boîte de messages : Les utilisateurs doivent pouvoir consulter les messages qui leur sont adressés par d'autres utilisateurs et stockés dans leurs boîte de messagerie. Vous êtes appelés à offrir :
  - plusieurs choix de classification pour l'affichage (triés selon la date, triés selon l'expéditeur)
  - plusieurs critères de recherche (date, expéditeur, objet, …)

- ✓ Stockage des Messages : Les messages envoyés et reçus doivent être stockés afin que les utilisateurs puissent accéder à leur historique de messages à tout moment.
  - Vous utiliserez des fichiers pour stocker les messages de manière permanente (un seul ou plusieurs selon votre choix)
  - Ces fichiers peuvent être sur la machine locale ou sur le serveur distant
  - Vous utiliserez les listes chaînées pour stocker et charger les messages dans la RAM.

- ✓ Suppression de Messages : Les utilisateurs doivent avoir la possibilité de supprimer les messages de leur boîte de réception une fois qu'ils les ont consultés.

Remarque *: On vous laisse le soin de choisir les propriétés qui décrivent les utilisateurs et les messages (nom, password, objet, nom_expéditeur, nom_destinataire, texte_message, ...) ou d'ajouter d'autres fonctionnalités plus raffinées que celles proposées.*

Pensez à créer un menu textuel quant au choix de l'utilisateur, voici un exemple seulement à titre indicatif :

Veuillez introduire votre choix :
Tapez : 1 – pour ajouter un utilisateur
2 – pour consulter la boîte de messagerie
3 – pour supprimer un message
...
0 – pour quitter l'application
...

Connexion au serveur de messagerie à distance (Optionnel) :

Vous pouvez enrichir votre application en offrant la possibilité de se connecter au serveur de messagerie à partir d'une machine distante.
Dans ce cas, vous devez utiliser ***les sockets*** et écrire le code client (qui tourne sur la machine distante) et le code serveur (qui tourne sur votre application).

Chiffrement des messages stockés (Optionnel) :

Les messages stockés peuvent être cryptés à l'aide d'algorithmes de cryptographie sécurisés pour assurer leur confidentialité.

3. Documents à fournir

Chaque étudiant (ou binôme) doit fournir :

1. Un **rapport** (sous forme pdf) qui comporte le(s) nom(s) de(s) l'étudiant(s) qui contient :
    - La description de la solution proposée : structures de données, algorithmes, structuration et modélisation des données
2. Les **codes source** du projet (**.c** et **.h**)
3. Une **démonstration vidéo** dans laquelle vous présentez les **résultats d'exécution** de votre programme.