

2013

Projet L3 STRI : Messagerie d'entreprise Client/Serveur

ALGORITHME

Ce document contient respectivement :

- L'algorithme de niveau 1 du fichier principal du Client : *mainClient.c*
- L'algorithme de niveau 1 du fichier principal du Serveur : *mainServeur.c*
- L'algorithme de niveau 2 du fichier principal du Client : *mainClient.c*
- L'algorithme de niveau 2 du fichier principal du Serveur: *mainServeur.c*



ALGORITHME DE NIVEAU 1 DU FICHIER PRINCIPAL DU CLIENT : *mainClient.c*

Le fichier mainClient.c est le fichier principal d'exécution du Client dans le cadre d'une messagerie d'entreprise Client/Serveur.

Une fois le fichier compilé puis exécuter, il permettra au client d'effectuer de nombreuses tâches proposées par la messagerie d'entreprise.

Via un menu d'authentification, il permet au client de choisir parmi les actions suivantes :

1-Création de comptes/Connexion à des comptes préexistants

2-Envoie de Mails

3-Consultation de sa boîte mail

1-Lecture des Mails

2-Suppression des Mails

En fonction du choix du numéro rentré par le client, il sera redirigé vers un menu adapté à sa requête.

En effet lors, de l'exécution du fichier mainClient.c, il sera afficher au client le menu d'authentification, lui proposant de se logger (le client doit saisir le chiffre 1), de créer un compte (le client doit saisir le chiffre 2) ou bien de quitter le programme (le client doit saisir le chiffre 3).

Ainsi :

- Dans le cas où le client choisit de se **logger**, il sera redirigé vers un menu principal lui proposant plusieurs tâches. Il aura le choix entre :
 - **Envoyer un mail** (le client doit saisir 1) : le client sera alors redirigé vers le menu d'envoi adapté : le login destinataire, l'objet ainsi que le texte du mail lui seront demandé.
 - **Consulter sa boîte aux lettres personnelle** (le client doit saisir 2) : la liste des mails présents dans sa boîte aux lettres sera alors affichée. Chacun de ses mails sera précédé d'un numéro. le client sera alors redirigé vers le menu de consultation adapté à sa requête. Plusieurs options lui sont proposées :
 - **Lire un mail** (le client doit saisir 1) : le client n'a plus qu'à choisir le numéro du mail qu'il souhaite consulté et ce mail lui sera afficher à l'écran.
 - **Supprimer un mail** (le client doit saisir 2) : le client n'a plus qu'à choisir le numéro du mail à supprimer et sa suppression sera faite automatiquement.
 - **Revenir au menu précédent** (le client doit saisir 3) : le retour vers le menu principal sera fait.
 - **Se déconnecter de sa session** (le client doit saisir 4)
 - **Connaitre le nombre de mails présents dans sa boîte aux lettres personnelles** (le client doit saisir 3) : ce nombre lui sera alors affiché à l'écran.
 - **Se déconnecter de sa session** (le client doit saisir 4).
- Dans le cas où le client souhaite **créer un compte**, il sera redirigé vers un menu lui demandant de saisir un login et un mot de passe personnel.
- Dans le cas où le client souhaite quitter le programme, une sortie de ce programme sera automatiquement réalisée par la machine (équivalent du Ctrl+C).

ALGORITHME DE NIVEAU 1 DU FICHIER PRINCIPAL DU SERVEUR : *mainServeur.c*

Le fichier *mainServeur.c* est le fichier principal d'exécution du Serveur dans le cadre d'une messagerie d'entreprise Client/Serveur.

Une fois le fichier *mainServeur.c* compilé puis exécuté, il permettra au serveur de répondre aux requêtes du client, de la façon la plus précise possible.

La réponse du serveur dépend alors du type de la requête que le client envoie.

Il va donc la lire et exécuter la tâche caractérisant cette requête.

Ainsi :

- S'il reçoit la requête **CREATE** *<login> <password>* de la part du client, il va lui créer un compte de login *login* et de mot de passe : *password*.
- S'il reçoit la requête **AUTHENTIFICATION** *<login> <password>* de la part du client, il va le connecter à la session ayant pour login *login* et de mot de passe : *password*.
- S'il reçoit la requête **HMANY** de la part du client, il lui retournera le nombre de mail présents dans la boîte aux lettres personnelles.
- S'il reçoit la requête **SEND** */<login_dest>/<objet>/<texte>* de la part du client, il stockera le fichier de nom : *login_objet* dans le répertoire personnel de la personne ayant pour login : *login_dest*.
- S'il reçoit la requête **LIST** de la part du client, il affichera la liste des mails présents dans sa boîte aux lettres personnelle.
- S'il reçoit la requête **DEL** *<n° mail>* de la part du client, il supprimera le mail ayant pour numéro *<n° mail>* de sa boîte aux lettres personnelle.
- S'il reçoit la requête **GET** *<n° mail>* de la part du client, il affichera le mail ayant pour numéro *<n° mail>* présent dans sa boîte aux lettres personnelle.

ALGORITHME DE NIVEAU 2 DU FICHIER PRINCIPAL DU CLIENT : *mainClient.c*

Tout d'abord, l'initialisation de la connexion avec le serveur doit être faite via la fonction Initialisation(char *machine), si ce n'est pas le cas, un code d'erreur est renvoyé .

Dans le cas où l'initialisation est faite avec succès :

- Le message de bienvenu est affiché.
- Le **menu d'authentification** s'affiche à l'écran du client, grâce à la fonction authentication_menu(); présente dans le fichier menu.c :
 - Si la variable « choice » prend comme valeur 1 (cas où le client souhaite se logger), la fonction identification_menu() est exécutée. Elle va alors afficher le **menu d'identification** à l'écran du client. Le client saisit alors son login et son password, il va alors envoyer au serveur une requête de type **AUTHENTIFICATION <login> <password>** qui va vérifier l'existence de ce compte. S'ils sont présents dans la base de données, le client accède à son compte, sinon un message d'erreur sera retourné au client. Tant que le **menu principal** va être affiché à l'écran du client (connexion réalisée avec succès), via la fonction main_menu() présente dans le fichier menu.c, la variable « choice » va à nouveau prendre une valeur, correspondant au numéro rentré par le client :
 - Si la variable « choice » prend 1 comme valeur (le client souhaite envoyer un mail), le client va être redirigé vers le menu d'envoi, via la fonction send_menu (char *request). Successivement, le client va saisir le login du destinataire, l'objet et le texte de son mail. Ces trois éléments vont être inscrite dans une requête de la forme : **SEND /<login_dest>/<objet>/<text>** que le client va envoyer au serveur. Au passage, le serveur vérifiera bien évidemment que login_dest saisie par le client existe bien dans la base de données.
 - Si la variable « choice » prend 2 comme valeur (le client souhaite consulter sa boîte aux lettres), le client envoie au serveur une requête de type **LIST** , qui va permettre d'afficher la liste des mails présents dans sa boîte aux lettres. De plus, un menu de consultation, via la fonction consultation_menu() va être affiché à la suite de cette liste de mail. L'utilisateur a alors le choix :
 - Il peut lire un de ses mails (la variable « choice » prend 1 comme valeur), il n'a qu'à saisir le numéro du mail qu'il souhaite lire. Il envoie ainsi au serveur une requête du type **GET <n° mail>**. Le mail choisit sera affiché à l'écran.
 - Il peut supprimer un de ses mails (la variable « choice » prend 2 comme valeur), il n'a qu'à saisir le numéro du mail qu'il souhaite supprimer. Il envoie ainsi au serveur une requête du type **DEL <n° mail>**. Le mail choisit sera alors supprimé.
 - Il peut revenir au menu précédent (la variable « choice » prend 3 comme valeur), le menu principal sera affiché de nouveau, via la fonction main_menu() ;
 - Il peut se déconnecter (la variable « choice » prend 4 comme valeur), il sera alors redirigé vers l'écran d'authentification (retour au tout début).

- Si la variable « choice » prend 3 comme valeur (le client souhaite connaître le nombre de mails présents dans sa boîte aux lettres), il envoie au serveur une requête de type **HMANY**, qui lui retournera ce nombre.
 - Si la variable « choice » prend 4 comme valeur (Il peut se déconnecter), il sera alors redirigé vers l'écran d'authentification (retour au tout début).
- Si la variable « choice » prend 2 comme valeur (cas où le client souhaite se créer un nouveau compte), le menu d'enregistrement va être affiché à l'écran du client via la fonction `registration_menu()` présente dans le fichier `menu.c`. Il va être demandé au client de rentrer un login et un mot de passe. Le client envoie alors au serveur une requête de type **CREATE <login> <password>**. Une vérification de l'existence du login va être effectuée du côté serveur. Si ce login n'existe pas, le compte va être créé, sinon un message d'erreur va être envoyé au client lui indiquant l'existence de ce login. Une nouvelle saisie auprès du client sera effectuée.
 - Si la variable « choice » prend 3 comme valeur (cas où le client souhaite quitter le programme), le programme prendra fin via la fonction `exit` (code d'erreur).

Une fois qu'un ou plusieurs de ces cas aient eut lieu, le client termine sa connexion avec le serveur via la fonction `Terminaison ()`.

ALGORITHME DE NIVEAU 2 DU FICHIER PRINCIPAL DU SERVEUR : *mainServeur.c*

Tout d'abord, le serveur doit initialiser sa connexion. Il va donc passer en mode connecté via la fonction Initialisation(), puis va attendre la connexion d'un client, via la fonction AttenteClient(). Quand le client se connecte, il va attendre la réception des différentes requêtes envoyées par ce dernier. Une fois la requête reçue, le serveur va devoir vérifier qu'elle ne soit pas nulle (si c'est le cas, le serveur termine la connexion avec le client via la fonction TerminaisonClient()), la lire, la mesurer (via la fonction strlen (char *message)) puis l'analyser.

- Si cette requête est de type "**CREATE**", le serveur va extraire le login et le mot de passe via la fonction extractlogs (requete, longueur_requete, login, password) suivant ce type. Il va alors exécuter la fonction int creation(char login[200], char password[50]) qui va permettre de créer un utilisateur et une MailBox avec le login et le password passés en paramètres, tout en vérifiant que le login saisi par l'utilisateur n'existe pas déjà.
- Si cette requête est de type "**AUTHENTIFICATION**", le serveur va extraire le login et le mot de passe via la fonction extractlogs (requete, longueur_requete, login, password) suivant ce type. Il va alors exécuter la fonction int authentification(char login[200], char password[50]) qui va permettre de vérifier si le login et le password passés en paramètres sont corrects, puis de connecter le client à son compte personnel.
- Si cette requête est de type "**HMANYN**", le serveur va exécuter la fonction int hmany (char *login) qui va récupérer le nombre de mails présent dans la MailBox de l'utilisateur (login) passé en paramètre. Il va l'envoyer au client dans un second temps.
- Si cette requête est de type "**SEND**", le serveur va extraire le login_dest, l'objet et le texte suivant le type. Il va ensuite exécuter la fonction int send(char* login_dest, char* login, char* object, char* message) qui a pour rôle de créer un fichier texte de nom « login objet » passées en paramètre et de le stocker dans la MailBox correspondant au login_dest.
- Si cette requête est de type "**LIST**", le serveur va afficher à l'utilisateur la liste des mails présents dans sa boîte aux lettres via la fonction list (char *login).
- Si cette requête est de type "**DEL**", le serveur va supprimer le mail de numéro <n° mail> (que la fonction extract_num (requete, longueur_requete) a pour but d'extraire de la requête) suivant le type. Il va pouvoir effectuer cette opération grâce à la fonction void deletion (char* login, int number) qui a pour but de prendre en paramètre le numéro du mail à supprimer et le login de l'utilisateur (pour supprimer le message de sa MailBox).
- Enfin si cette requête est du type "**GET**", le serveur va afficher à l'écran le mail de numéro <n° mail>. Il faut alors extraire ce numéro de la requête via la fonction extract_num (requete, longueur_requete). La fonction void read (int num, char *login) permet la lecture du numéro de mail passé en entrée de la MailBox de l'utilisateur (login) passé en paramètre.

A la suite d'un de ses cas, la requête reçue est vidée (free (request)) afin de renouveler sa réception avec l'un des types spécifiques présentés ci-dessus.

Ce déroulement à lieu jusqu'à ce que le client n'émette plus aucune requête, ainsi la connexion Client/Serveur serait coupée.