

BARRY

HAKIM

BECKER

PROJET Client/Serveur STRI L3

Alogo client

premier niveau de raffinage

debut:

test si le client est lancé avec les bons parametres

si pas ok

 ecrire(message d'erreur)

si ok

si (initialisationAvecServeur()!=1)

 ecire(error initialisation)

sinon

 affiche Menu() et attend une saisie(choix)

 si choix=1

 appel send_email()

 si choix=2

 appel login()

 si choix=3

 appel signup()

 si choix=4

 sort du programme

 sinon

 ecrire(je ne comprend pas votre choix)

 termine la connection

fin:

ALGORITHME SERVEUR

1^{er} niveau de Raffinage.

Initialisation du serveur :

-Vérifie si le serveur à été lancé avec un numéro de port. Si ce n'est pas le cas une erreur est retournée de type « server no_port ».

-Initialisation du serveur et on vérifie si aucune erreur s'est produite. S'il y a une erreur d'initialisation une erreur est retournée de type « Erreur d'initialisation »).

Connexion au serveur :

(Fonction nommée : AttenteClient()).

-On attend qu'un client se connecte au serveur

 *Vérification de la connexion Client/Serveur.

 **Si la connexion est non valide une erreur est

retournée de type « AttenteClient, erreur de accept ».

 **Sinon on enregistre les informations du client puis on

retourne un message de confirmation de connexion : « Client sur la machine d'adresse X connecté ».

 **Sinon si il n'y a pas d'information concernant le client, ce

dernier est connecté en Mode Anonyme, et on retourne un message de confirmation de connexion en

Mode anonyme : « Client anonyme connecté ».

Communication Client/Serveur :

(Fonction nommée : dostuff()).

-Vérifie le type de la requête du client :

*Si c'est une requête de type « login » alors l'utilisateur va devoir entrer des identifiant (ID & Password) afin de se connecter à sa boîte mail.

**Si l'identification est validée le système propose trois options :

***Envoyer un mail au destinataire souhaité.

***Lire un mail de sa boîte mail.

***Effacer un mail de sa boîte mail.

*Si la requête est de type « signup » alors l'utilisateur pourra créer sa boîte mail.

Terminaison Client :

(Fonction nommée : TerminaisonClient()).

-Termine la communication du client.

2^{ème} niveau de Raffinage.

la fonction send_email()

send_email()

debut:

message caractère
input tableau[80] caractère
mes tableau[256] caractère
structure email
buffer tableau[5888] caractere

si(emission("send\n")!=1)
ecrire(erreur d'emission)

message=reception()
si(compare(message et ok)!=0)
ecrire(erreur d'emission)
ecrire(separer email par ';' \n \n)
ecrire(\n to:)
lire(email.to)
ecrire("\ncc: ")
lire(email.cc)
ecrire("\nbcc: ")
lire(email.bcc)
ecrire("\nsubject: ")
lire(email.subject)
ecrire("\nmessage: ")
lire(email.message)

```

        stoque email dans buffer
        Emission(buffer)
    fin
}

```

la fonction signup()

```

void signup()
debut:
    message = NULL pointeur
    lname, password, passwordconf tableau[80] caractère
    mes tableau [256] caractère
    i entier

    si(Emission("signup\n")!=1)
        ecrire("Erreur d'emission\n")

    message = Reception();
    ecrire(message);

    si(compare(message,"ok\n")!=0)
        ecrire("Erreur d'emission\n")

    ecrire("*** Creating a new account ***\n\n")

    ecrire("Last name: ")
    lire(lname)

    ecrire("\n\nCreate a Password: ")
    lire(password)

    ecrire("\n\nConfirm your Password: ")
    lire(passwordconf)

    si(compare(password,passwordconf) == 0)
    debutsi:

        copie lname dans mes
        introduit dans mes un espace
        copie password dans mes
        ajoute \n dans mes

        si(Emission(mes)!=1)
            ecrire("Erreur d'emission\n")

        libere contenu(message)
        message = Reception()

        ecrire(message)
    finsi:
    sinon

```

```

    debutsinon:
        ecrire("\n\n*** The password and the password confirmation doesn't match
***\n\n\n");

    finsinon
        libere contenu(message)
fin

void login()
debut:
    message = NULL pointeur
    input tableau[80] caractère
    mes tableau [256] caractère
    i entier

    si(Emission("login\n")!=1)
        ecrire("Erreur d'emission\n")

    ecrire(message)
    message = Reception();
    ecrire(message);

    si(compare(message,"ok\n")!=0)
        ecrire("Erreur d'emission\n")

    ecrire("***** Signing in *****\n\n")
    ecrire("Email: ")
    lire(input)
    copie input dans mes
    ecrire("Password: ")
    libere contenu(input)
    lire(input)
    introduit un espace dans mes
    copie input dans mes
    ajoute \n dans mes

    si(Emission(mes)!=1)
        ecrire("Erreur d'emission\n")

    ecrire(message)
    libere contenu(message)
    message = Reception()
    ecrire(message)

/*else correct password connection established*/

tant que((i!=1)&&(i!=2)&&(i!=3)&&(i!=4))
debutTQ:
    ecrire("%s", message);
    ecrire("\n\nWhat would you like to do:\n\n-Read an email?\t\t---> press 1 \n-Delete
an email?\t---> press 2 \n-Compose a new mail?\t---> press 3 \n-Signout?\t\t---> press 4\n\n")

```

```

lire(i);

si(i==1)
    read_email();

sinon si(i==2)
    delete_email();

sinon si(i==3)
    send_email();

sinon si(i==4)
    sort de la fonction

sinon
    ecire("\nI can not understand your choice\n\n");

finTQ:
libere contenu(message);
fin:

```

SERVEUR :

Initialisation du serveur :

-Vérifie si le serveur à été lancé avec un numéro de port. Si ce n'est pas le cas une erreur est retournée de type « server no_port ». (Raffinage maximum)

-Initialisation du serveur :

*Si une mauvaise adresse de port est utilisée lors de l'initialisation un code d'erreur est retourné de type : « Initialisation, erreur de getaddrinfo XXX ».

*S'il y a un problème lors de l'utilisation du port, une erreur est retournée de type : « Initialisation, erreur de bind ».

*Si l'initialisation c'est correctement déroulée, un code de retour valide est retourné.

Connexion au serveur :

(Fonction nommée : AttenteClient()).

-On attend qu'un client se connecte au serveur

*Vérification de la connexion Client/Serveur.

**Si la connexion est non valide, la fonction « Accept » retourne un code de retour -1

**Sinon les informations du client sont utilisées via la fonction « getnameinfo() ».

**Sinon si il n'y a pas d'information concernant le client, la fonction « getnameinfo() » retourne un nombre différent de 0.

Communication Client/Serveur :

(Fonction nommée : dostuff()).

-Vérifie le type de la requête du client par comparaison avec la chaîne de caractère reçu du client avec les chaînes de caractères login et signup:

*Si c'est une requête de type « login » alors l'utilisateur va devoir entrer des identifiants (ID & Password) afin de se connecter à sa boîte mail.

**On récupère l'adresse client « ad » ainsi que le mot de passe « ps » pour les comparer avec la liste des utilisateurs autorisés via la fonction « checkmail() ».

***Si l'identificateur n'est pas identifié un code de retour est retourné de type : « 306 »

***Si l'identification est validée le système propose trois options :

****Envoyer un mail au destinataire souhaité.

****Lire un mail de sa boîte mail.

//Lecture d'un mail :

->Emission sur le Client de tous les mails de sa boîte mail. Chaque mail est décrit par son Nom d'Objet.

->Le Client envoie le numéro du mail qu'il veut lire, sinon il envoie '0'.

->On compare la chaîne de caractère envoyée avec le chiffre '0', si positif on retourne au menu parent sinon on donne en paramètre de la fonction « read_email() » le chiffre donné.

->La fonction « read_email() » Affiche sur le client le mail choisi.

****Effacer un mail de sa boîte mail.

//Effacer un mail

->Emission sur le Client de tous les mails de sa boîte mail. Chaque mail est décrit par son Nom d'Objet.

->Le Client envoie le numéro du mail qu'il veut effacer, sinon il envoie '0'.

->On compare la chaîne de caractère envoyée avec le chiffre '0', si positif on retourne au menu parent sinon on donne en paramètre de la fonction « move_email() » le chiffre donné.

->La fonction « move_email() » efface le mail sélectionné.

*Si la requête est de type « signup » alors l'utilisateur pourra créer sa boîte mail via l'identifiant et le mot de passe qu'il aura donné.

*On lit la requête contenant les informations du client puis on les stocke dans les chaînes de caractères « lname » et « password ».

*On ouvre le fichier « list.txt », qui est le fichier des utilisateurs inscrits.

*Si il y a une erreur fichier une erreur est retournée du type : « The account can't be created due to an internal error ».

*Sinon on inscrit l'utilisateur dans le fichier

*On envoie une confirmation de création du compte en lui affichant son adresse email complète de type : « The account XXX@company.net has been successfully created »

*Si le compte existe déjà on retourne un message de non validation tel que : « The account XXX@company.net already exists, try something else ».

Terminaison Client :

(Fonction nommée : TerminaisonClient()).

-Termine la communication du client.