



Réseaux pour ingénieurs GLO-2000

TP4 : Serveur de courriel

Professeur responsable:

Ronald Beaubrun

Ronald.Beaubrun@ift.ulaval.ca

Responsables des travaux pratiques:

Louis Fortier-Dubois et Jean-Alexandre Beaumont

glo2000a17@gmail.com

Mise en situation

Vous avez été mandaté par une compagnie afin de leur concevoir un système de courriel interne (@reseauglo.ca). Le système doit être composé d'un client et d'un serveur. Le serveur sera un relais SMTP qui traitera les envois de courriel à l'externe et à l'interne. Les utilisateurs utiliseront le client afin d'envoyer et consulter leurs courriels.

Fonctionnement du programme

Voici un scénario d'utilisation de la solution (1 point) :

- Lors du démarrage du client, un premier menu est affiché à l'utilisateur :

```
Menu de connexion
1. Se connecter
2. Creer un compte
```

- L'utilisateur doit choisir une option en entrant un nombre proposé dans le menu. Dans l'un ou l'autre des cas (voir les scénarios de connexion et de création de compte pour plus de détails), le client entre son nom d'usager et son mot de passe dans le client, lequel transmet ces informations au serveur. Le serveur indique au client si la connexion est acceptée ou refusée.
- Si la connexion est refusée, un message d'erreur est affiché à l'utilisateur.
- Si la connexion est acceptée, un second menu est affiché à l'utilisateur :

```
Menu principal
1. Envoi de courriels
2. Consultation de courriels
3. Statistiques
4. Quitter
```

- Le menu doit être de nouveau disponible pour l'utilisateur après chaque action sauf le mode 4.

Voici un scénario de connexion (1 point) :

- Une fois le nom d'usager et le mot de passe entrés par l'utilisateur, le client envoie ces informations au serveur.
- Le serveur vérifie que l'utilisateur existe. S'il existe, il hache le mot de passe (voir section *Utilitaires*) reçu et vérifie qu'il correspond au mot de passe haché stocké dans le fichier « config.txt » de l'utilisateur.
- Le serveur signale au client si tout est conforme ou pas.

Voici un scénario de création de compte (1 point) :

- Une fois le nom d'usager et le mot de passe entrés par l'utilisateur, le client envoie ces informations au serveur.
- Le serveur vérifie que l'utilisateur n'existe pas déjà. Le cas échéant, il s'assure que le mot de passe contient entre 6 et 12 caractères, dont au moins un chiffre et une lettre (utilisez une regex!), puis le hache. Il crée ensuite un répertoire au nom de l'utilisateur, dans lequel il crée un fichier « config.txt » contenant le mot de passe haché sur la première ligne.
- Le serveur signale au client si tout est conforme ou pas.

Voici un scénario d'envoi de courriel : (2 points)

- Le client demande à l'utilisateur l'adresse de destination, le sujet et le corps du message. Ces informations sont envoyées au serveur.
- Si la destination est une adresse interne (finissant par @reseauglo.ca), le message est déposé dans le dossier de l'utilisateur correspondant. Si l'utilisateur n'existe pas, le message est déposé dans le dossier « DESTERREUR ».
- Si la destination est une adresse externe, le message est envoyé par SMTP à l'aide du serveur « smtp.ulaval.ca ».
- Le serveur envoie une confirmation au client.
- Le client retourne au menu.

Voici un scénario de consultation de courriel : (2 points)

- Le serveur envoie au client une liste numérotée des sujets des courriels de l'utilisateur.
- Le client demande un numéro à l'utilisateur.
- Le client envoie le numéro entré au serveur. Celui-ci retourne au client le message correspondant.
- Le message est affiché et le client attend une entrée de l'utilisateur puis retourne au menu.

Voici un scénario d'affichage des statistiques : (1 point)

- Le client affiche le nombre de messages contenu dans le dossier de l'utilisateur, la taille totale de son dossier (voir section *Utilitaires*) et finalement une liste des messages par sujet.
- Le client attend une entrée de l'utilisateur puis retourne au menu.

Éléments que le programme doit être en mesure d'accomplir

- Les dossiers des utilisateurs doivent être situés dans le même répertoire que le serveur. (0.5 point)
- Le dossier d'un utilisateur doit contenir un fichier de configuration « config.txt » contenant sur la première ligne le mot de passe (haché) de l'utilisateur. (0.5 point)
- Les mots de passe doivent être hachés dans le fichier de configuration. (0.5 point)
- L'utilisateur doit s'authentifier correctement avant de pouvoir effectuer d'autres opérations. (1 point)
- Le client ne doit pas afficher le mot de passe lorsque celui-ci est écrit (voir section *Utilitaires*). (0.5 point)
- Le serveur et le client doivent traiter les erreurs sans arrêt de l'exécution. (1 point)
- Le client doit demander les informations d'envoi de courriel à l'utilisateur puis les transmettre au serveur. (1 point)
- Le serveur doit forcer les nouveaux mots de passe à contenir 6 à 12 caractères dont au moins un chiffre et une lettre. (1 point)
- Lors de la connexion, le serveur doit refuser un nom d'utilisateur absent, tandis que lors de la création de compte, il doit refuser un nom d'utilisateur déjà présent. (1 point)
- Le serveur doit valider les informations d'envoi puis transmettre la confirmation au client (succès ou échec). (0.5 point)
- Après la création de compte, le menu principal doit être affiché directement, comme si l'utilisateur venait de se connecter de façon standard. (0.5 point)

- Dans le cas d'un courriel à l'externe, le message est envoyé via *smtp.ulaval.ca*. (1 point)
- Dans le cas d'un courriel finissant par « @reseauglo.ca », le message est stocké dans le dossier de l'utilisateur ou dans « DESTERREUR » s'il n'existe pas. (1 point)
- L'affichage des statistiques doit inclure le nombre de messages, la taille du dossier ainsi que la liste des sujets des messages reçus. (1 point)
- Le client doit quitter correctement. (1 point)

ATTENTION, le client n'accède pas directement aux informations. Dans tous les cas, c'est le serveur qui envoie les informations au client. C'est-à-dire que le client pourrait être exécuté sur une autre station que celle où les informations sont stockées.

Utilitaires

- Hacher un texte en python :

```
from hashlib import sha256

plaintext = input("Texte à hacher : ")
hashedText = sha256(plaintext.encode()).hexdigest()
print("Texte hache : " + hashedText)
```

- Obtenir la taille d'un fichier en python :

```
from os.path import getsize

filename = input("Nom du fichier : ")
size = getsize(filename)
print("Taille en octets : " + str(size))
```

- Faire entrer un mot de passe par l'utilisateur sans l'afficher :

```
import getpass

password = getpass.getpass('Password : ')
```

Modalités de remise

- À faire en équipe de 2 ou 3.
- Voici ce que vous devez remettre :
 - Le code source du serveur (doit s'appeler exactement "serveur.py")
 - Le code source du client (doit s'appeler exactement "client.py")
 - Tout autre fichier utilitaire requis par votre application
- **Vos menus doivent correspondre exactement avec les exemples donnés dans ce document.**
- Remettre sur le portail des cours.
- Date limite : 6 décembre
- Tout travail remis en retard se verra attribuer la note 0.