

A4 2cm1.5cm2cm1.5cm

Rapport de PR

Ayrault Maxime 3203694
Caristan Mathis 30

03/01/2017

Introduction

Notre programme prend en entrée un numero de port *num_port*,
un nombre de client *num_client* max en parallele
et un nombre de requetes limites par 10 secondes *num_cpt*.

1 Question 1 Structure du serveur (fait) :

1.1 Structure Client

```
typedef struct {
    /* contient les information sur la socket */
    struct sockaddr_in csinf;
    /* le descripteur sur la socket du client */
    int sock;
    /* le thread qui lui est attach\'e */
    pthread_t thread;
    /* le numero d\'index du client dans free_client */
    int index;
    Loginfo loginfo;
} Client;
```

On a commencé par créer un serveur web TCP avec une socket dont le numéro de port est donnée en entrée du programme.
structure interne du serveur :

fd : un numero de descripteur de socket.

sinf : structure contenant les informations sur la socket du serveur.

free_client : une bitmap de taille *num_client* pour les clients libre

clients : un tableau de structure struct Client qui contient les informations liées chaque client présent sur le serveur

cpt : le nombre de clients l\'instant T sur le serveur

? : est ce que j\'en ai oublié

On distingue ensuite deux cas :

- *?* si une entrée clavier est détectée
- *?* si une entrée sur la socket est détectée

Si une entrée clavier est détectée:

on regarde si cela correspond QUIT ou quit
— ¿ si oui on ferme le serveur
— ¿ sinon on ne fait rien

Si une entrée sur la socket est détectée:
on cherche une place dans un tableau de taille *taille* donné
comme
argument en entrée du programme.
– ¿ Si il y en a une :
— ¿ on tente d'accepter la nouvelle connexion pour
recupérer
un nouveau descripteur sur la nouvelle socket
— ¿ on remplit notre structure Client avec toutes
les
informations
de la nouvelle socket
— ¿ on lance un nouveau thread qui s'occupera
de traiter la
requête reçue avec la structure contenant les informations
comme
argument
– ¿ Si il n'y en a pas
on ferme la connexion en renvoyant une page html
avec le
bon header

Le traitement fait dans le thread :

- ¿ la vérification de la requête bien formée
- ¿ la recherche du type mime
- ¿ vérification si ouvrable ou pas avec stat
- ¿ vérification des droits sur le fichier
- ¿ ouverture du fichier
- ¿ envoi du contenu du fichier au client
- ¿ vider la structure pour permettre de nouvelles connexions

- ζ fermer la connexion et finir le thread

2 *Question 2 Journalisation (fait) :*

recuperer les informations du client et les stocker dans notre structure

puis toute les ecrire a la fin dun fichier /tmp/httpnumetu

3 Question 3 Fichier executable (fait) :

dans la verification des droits du fichier regarder si le bit exe est mis

1 et le stocker dans une variable exe

mettre un flag dessus fans le test de quoi faire apres si le fichier est executable :

– ¿ creer deux tube nomée: les noms sont synchro tid et retour tid

— ¿ un pour la syncho

— ¿ un pour renvoyer de que renvoie l'execution de l'executable

– ¿ faire un fork

— ¿ dans le fils

— ¿ ouvrir les deux fifo une en lecture l'autre en écriture

— ¿ rediriger la sortie standard vers la fifo en écriture

— ¿ attendre le feu vert envoyé par le pere pour commencer

— ¿ lancer un execl avec le fichier executable

— ¿ dans le pere

— ¿ ouvrir les deux fifo dans le sens inverse du fils

(ouverture - ¿ écriture)

— ¿ envoyer le feu vert au fils

— ¿ renvoyer au client tout ce que le fils lui a fait

parvenir par le tube

– ¿ unlink des tubes

sinon

– ¿ faire la même chose qu'avant

4 *Question 4 Requetes persistentes (fait) :*

modification de la fonction testant si la requete est bien formé
pour renvoyer aussi si la requette correspond a une requete en
http1.0 ou en http 1.1

mettre comme condition darret du traitement des requetes reu
avant de fermer la connexion le type de requette reçu

- ¿si en http1.0 faire un seul tour de boucle puis fermer
la connexion
- ¿ si en http1.1 faire le traitement de chaque requettes
reu jusqu'a n'avoir plus rien a lire

aure truc que j'ai pas forcément compris aussi...

5 Question 5 Contrer le déni de services (pas fait, manque de temps) :

gestion de la question :

- ¿ si timer arrive a 0 :
mettre le nombre de visite de l'ip -1
- ¿ avoir un tableau 2 dimensions de taille fixe 5 ou 10 fois
plus grand que le nombre de clients simultanés
contenant l'adresse IP du client et le nombre de visite

réalisation :

chaque requête reçue :

- ¿ lancer le timer
- ¿ checker si dans le tableau des ip reçues
 - ¿ si oui faire +1
 - ¿ regarder si nombre bon ou pas
 - ¿ si bon exécuter la requête
 - ¿ si pas bon jeter la requête
 - ¿ si = -1 mettre un dans le tableau et exécuter la requête

problème :

- ¿ nécessite un énorme tableau
- ¿ de le vider de temps en temps aussi pour gagner du temps dans le parse du tableau
- ¿ ou mettre -1 dans la valeur de l'ip ou du nombre