Rapport de Projet : Un serveur HTTP

Cai Qiaoshan 3605744 travail personnelle

1. Structure du serveur

le processus principale crée socket TCP dans le domain Internet, modifie des options pour autoriser de réutiliser une address. Quand le client se connecte, le processus principal créé une thread et passe le descripteur de la socket, on n'attend pas la fin de cette thread. La thread récupérer le chemin de fichier le clients demande et analyser l'existence et droit d'access. Je crée les fonctions « existpas » et « forbiden » qui vont être appelé par la thread dans la situation correspond. Chaque fonction va retourner une message avec 3 chiffre et écrit le contenu de cette fichier dans le descripteur de la clients. Je créé aussi un fonction « fichiertype » pour trouver le type de fichier.

2. Journalisation

Je ajout une fonction « addlog » pour créer une fichier de log. méthode getpeername permet de trouver struct sockaddr correspond à un descripteur socket et méthode inet_ntop renvoyer l'address IP sous forme de string. Méthode gettimeofday permet d'obtenir le temps. Quand cette fonction crée ou écrit la fichier log tous les information demandé le mutex est utilisé pour protéger ce fichier log commun.

3. Fichier exécutable

Je crée une fonction « execute » qui va être appelée si la fichier demandé a le droit d'executer. Dans cette fonction, fork() est utilisé pour créer une processus fils. Avant créer le fils, les signaux SIGINT et SIGQUIT est ignoré parce que le processus père ne doit pas être affecter par les deux signaux pendant execution de son fils. Dans le processus fils, il doit remettre le comportement de SIGINT et SIGQUIT comme défaut et puis execute le commande. Le père attend la fin de son fils, je crée une boucle et utiliser la valeur retour de waitpid() comme condition de sortir la boucle. Si la boucle ne fini pas pendant 10 seconds le père vas envoyer le SIGQUIT pour tuer le fils.

4. Requêtes persistentes

Je modifie le processus principale que au lieu de crée une thread pour chaque clients, il va crée une thread pour chaque demande de la clients. Chaque thread va réaliser les travails comme dans les question précédents. Les commendes HTTP sont séparé par des ligne vides. Méthode setlinebuf permet de lire une fichier ligne par ligne. Je crée une boucle pour ignorer les lignes vides et récupérer tous les demandes.

5. Contrer le déni de services

Chaque fois une clients se connecte, le processus principale va crée une thread, les autre threads pour les demandes de clients lui envoient la taille de fichier. Si la taille de fichier dépasse une seuil qui passe comme troisième argument. La nouvelle thread va mettre un flag global à 0 et le serveur va examiner ce flag dans son boucle et il va ignorer cette clients pendants 10 seconds.