

LO14 1er rapport

Le projet LO14 a pour objectif de créer un **serveur d'archive**. Il s'agit de créer une **nouvelle commande permettant de communiquer** avec ce serveur. Ce serveur doit être **disponible en permanence** et doit **pouvoir communiquer** au travers de différentes commandes entrées par l'utilisateur. Le script utilisateur et serveur utilisent l'outil **Netcat**. Ils communiquent avec un tube qui sera défini par le serveur. Le tube sera **détruit** lorsque le script côté client sera interrompu ou terminé. De plus, nous voulons que **plusieurs utilisateurs** puissent se connecter simultanément au serveur. Il faudra donc **attribuer un tube** pour chaque client se connectant au serveur. Le serveur fonctionne en mode **listening**. Il attend les requêtes client. Afin de travailler de manière organisée, un GitHub est créé pour synchroniser et éditer différentes versions du projet. Discord et Facebook Messenger sont également employés pour communiquer et échanger des idées.

Les commandes tapées doivent avoir les caractéristiques suivantes et respecter une certaine syntaxe :

➤ Mode list

Afficher la liste des archives (grâce à une commande **"ls" côté serveur**) présente sur le serveur. Un test est effectué pour savoir si des archives existent ou non dans le serveur. Si il y a existence, le serveur affiche une réponse pour avertir l'utilisateur. Sinon, il répond qu'il n'y a aucune archives présentes.

➤ Mode create

Créer une archive portant comme nom l'argument écrit par le client lors de l'appel de la fonction avec l'arborescence du répertoire courant du client.

Attention ! Notre fonction (côté client) devra donc être récursive pour pouvoir prendre en compte les répertoires sous-jacents.

Les commandes dont ont aura besoin :

→ **ls -la (côté client)**

Des pistes pour la création d'archives :

- Création de l'archive côté client et transmission au serveur qui écrit dans un fichier.
- Transmission des **résultats des "ls -la"** du client bout par bout au serveur avec un message de fin de transmission : le serveur traite donc les données bout par bout (peut-être compliqué pour synchroniser les deux parties).
- Transmission de la **concaténation de tous les "ls -la"** réalisés au serveur, et traitement unique du serveur.

Utilisation de **awk, grep, sed, cut** (traitement de données)

➤ Mode Browse

Entrer dans une archive, **naviguer** dedans et **réaliser des opérations** sur l'archive

pwd

Garder en mémoire dans le script serveur le répertoire courant grâce à une variable globale. Il suffit donc de transmettre la variable globale au client qu'il affichera.

ls

Liste les fichiers et les dossiers courants de l'archive. Les paramètres **-l -a et -la** seront acceptés et l'on pourra spécifier un chemin vers un dossier spécifique de l'archive en argument. Pour le traitement de données présentes dans l'archive, l'utilisation de **awk** sera privilégiée.

cd

Actualisation de la variable globale du répertoire courant : le client envoie en argument de **"cd"** le chemin du dossier destination.

cat

Affiche le **contenu de maximum deux fichiers** donc le chemin a été fourni en argument.

Utilisation de **"grep"** et **"sed"** côté serveur

rm

Suppression du fichier ou du dossier dont le chemin est fourni en argument. Attention ! La fonction devra être récursive pour supprimer d'éventuels dossiers sous-jacents.

Utilisation d'une **boucle for** dans le script serveur avec un compteur de ligne **supplémentaire et "sed" pour supprimer les lignes de l'archive souhaitées plus "grep"** pour obtenir les lignes des données d'un fichier à supprimer **ou awk + sed**.

touch

Permet de **créer une entrée dans l'archive** pour un fichier vide dont le nom est fourni en argument (ou chemin) si le fichier n'existe pas déjà.

Utilisation de **awk** pour ajouter la nouvelle ligne concernant le fichier dans son dossier de l'archive.

mkdir

Création d'un répertoire ou de répertoires avec l'**option "-p"**.

Utilisation de **awk** pour créer les entrées dans l'archive pour chaque répertoire créé.

➤ Mode Extract

Extraire le contenu d'une archive dont le nom est renseigné en argument vers le répertoire courant en **respectant bien les droits d'accès** pour les fichiers ou les dossiers.

Le **serveur transmet le contenu complet de l'archive au client**, qui en extrait les données et crée les fichiers et les dossiers correspondants.

Utilisation d'une boucle for pour parcourir l'archive et création des fichiers et dossiers correspondant selon la présence de 'd' ou '-' avant les droits (grep), application des droits à l'aide de "**chmod**" et "**grep**" (sélections des droits), puis sélection des lignes de données (si il y en a) à l'aide de "**grep**" et récupération du contenu des lignes avec "**sed**" **ou** utilisation de **awk**.