

LINFO1252 – Systèmes informatiques

Travail Dirigé (TD) S1 : Introduction au shell et aux scripts Bash

Objectifs de ce TD : Dans ce TD, vous allez créer un script Bash qui permet de compresser un répertoire et de générer un fichier journal (log) avec des informations détaillées sur les fichiers de ce répertoire. Le script attendu aura pour fonctionnalités :

1. **Sauvegarde de fichiers :** Compresser un répertoire donné dans un fichier `.tar.gz`.
2. **Parcours récursif des dossiers :** Identifier tous les sous-répertoires du répertoire sauvegardé.
3. **Extraction des tailles des fichiers :** Lister les tailles des fichiers par ordre décroissant dans un log file.
4. **Gestion des erreurs :** Vérifier la validité des entrées (arguments) et des répertoires avant d'exécuter le script.

À la fin du TD, vous aurez un script qui prendra en entrée deux arguments : un répertoire à sauvegarder et un nom d'archive. Le script générera un fichier log détaillant la taille de chaque fichier dans le répertoire sauvegardé, trié par ordre décroissant de taille. Ce fichier se trouvera dans le même répertoire que l'archive de sauvegarde.

Compétences travaillées : Méthodologie de conception d'un script Bash et maîtrise des outils du terminal.

Prérequis : Ce TD nécessite un accès à un shell.

Étape 1 : Initialisation du script Bash (5 minutes)

Créez un fichier nommé `backup.sh` et commencez par ajouter la ligne shebang qui indique l'interpréteur Bash à utiliser. Le shebang, représenté par `#!/`, est un en-tête d'un fichier texte qui sert à indiquer que ce fichier est un script. Il est suivi par l'interpréteur permettant d'exécuter ce script. Ajoutez ensuite une commande simple pour afficher un message. Testez votre script dans le terminal.

Étape 2 : Variables, compression et journalisation (15 minutes)

Remplacez la commande simple de votre script par une suite de commandes qui permet de :

- Compresser un dossier à sauvegarder. Le chemin de celui-ci est écrit en dur dans le script
- Créer un fichier de log vide
- Écrire dans ce fichier de log la date et l'heure du backup

Une fois que vous arrivez à faire fonctionner ces commandes, placez leurs arguments dans des variables Bash pour pouvoir vous en servir plus loin dans le script.

Étape 3 : Construction de la commande pour lister les dossiers (10 minutes)

Dans cette étape, vous allez explorer la commande `find` pour obtenir une liste des répertoires de manière structurée. Parcourez la manpage pour comprendre le fonctionnement de `find` et trouver quels arguments lui fournir.

Étape 4 : Extraction des tailles des fichiers (20 minutes)

L'objectif ici est de lister les fichiers dans un dossier et d'extraire leur taille. Nous allons construire progressivement cette commande. Vous devez pour ça :

- Essayer d'afficher (notamment) la taille des fichiers dans un dossier (cherchez dans la manpage de `ls`)
- Redirigez la sortie de `ls` pour être traitée dans `awk`. Dans un affichage en colonnes, la commande `awk '{print $1, $3}'` va filtrer pour n'afficher que la première et la troisième colonne.
- (Optionnel) Utilisez la commande `sed` pour que le chemin jusqu'au fichier soit juxtaposé devant le nom du fichier.
- Redirigez le tout dans un fichier temporaire

Étape 5 : Implémenter la récursion et finaliser la journalisation (25 minutes)

Maintenant, vous allez parcourir tous les dossiers obtenus à l'étape 3 et exécuter la commande de l'étape 4 pour chaque dossier afin de récupérer les informations de taille des fichiers. Pour cela, **itérez** sur la liste de dossier de l'étape 3.

Quand vous avez la liste de tous les fichiers du répertoire à sauvegarder et de leurs tailles, triez-la en fonction des tailles.

Étape 6 : Gestion des arguments et des conditions (15 minutes)

Pour terminer, vous allez ajouter des vérifications dans le script pour gérer les arguments et vérifier l'existence des fichiers.

- Vérifier si le répertoire à sauvegarder existe
- Vérifier si un nom d'archive est fourni
- Assignez aux variables que vous utilisez les valeurs fournies en argument