

TP : Gestion d'une Bibliothèque avec des Scripts Bash

Objectif : Au travers de ce TP, nous allons créer un script Bash qui permet de gérer une petite bibliothèque. À chaque étape, le script sera enrichi avec de nouvelles fonctionnalités.

Rendu :

- Format PDF
- milodigitalcreations@gmail.com
- Nom du fichier : *PRENOM-NOM-BSSR3-SBASH.pdf*
- A chaque étape, me donner le script ET le résultat de son exécution (capture d'écran ou copier/coller)

Pré-requis :

- Avoir un environnement Linux.
- Avoir complété le TP guidé d'introduction.
- Ouvrir un terminal.
- Créer un dossier pour ce TP.

Étape 1 : Initialisation du Script et de la Base de Données

1. Créez un fichier ``bibliotheque.sh``.
2. Ajoutez le shebang en début de fichier : ``#!/bin/bash``.
3. Créez un fichier ``livres.txt`` qui servira de base de données.

Étape 2 : Ajout d'un Livre

Le script doit permettre d'ajouter un livre avec son titre et son auteur.

1. Ajoutez la possibilité d'ajouter un livre en utilisant la commande : ``./bibliotheque.sh ajouter "Titre du Livre" "Auteur"``.
2. Le livre doit être ajouté à ``livres.txt`` sous le format : ``Titre du Livre - Auteur``.

Étape 3 : Listage des Livres

Ajoutez une option pour lister tous les livres.

1. En utilisant la commande `./bibliotheque.sh lister`, tous les livres doivent être affichés.
2. Utilisez une boucle `while` pour lire chaque ligne du fichier.

Étape 4 : Recherche de Livre

1. Ajoutez une option `rechercher` qui prend un mot-clé en paramètre.
2. La commande `./bibliotheque.sh rechercher "Titre"` affichera tous les livres contenant le mot-clé.

Étape 5 : Suppression d'un Livre

1. Ajoutez une option `supprimer` qui prend un titre en paramètre.
2. La commande `./bibliotheque.sh supprimer "Titre du Livre"` supprimera le livre correspondant de `livres.txt`.

Étape 6 : Ajout de Paramètres Positionnels

1. Modifiez votre script pour qu'il accepte des paramètres positionnels.
2. Par exemple, `./bibliotheque.sh ajouter` devrait demander interactivement le titre et l'auteur.

Étape 7 : Gestion d'Erreurs

1. Si le fichier `livres.txt` n'existe pas, le script doit afficher un message d'erreur et quitter.
2. Si une action inconnue est demandée, affichez un message d'aide.

Étape 8 : Rendre le Script Interactif

1. Si le script est lancé sans arguments, proposez un menu interactif.
2. Le menu doit offrir des options pour ajouter, lister, rechercher et supprimer des livres.

Étape 9 : Sauvegarde de la Bibliothèque

1. Ajoutez une option `sauvegarder` qui crée une copie de sauvegarde de `livres.txt`.
2. La commande `./bibliotheque.sh sauvegarder` créera un fichier `livres_backup.txt`.

Étape 10 : Restauration de la Bibliothèque

1. Ajoutez une option `restaurer` qui restaure `livres.txt` depuis `livres_backup.txt`.
2. La commande `./bibliotheque.sh restaurer` remplacera `livres.txt` par `livres_backup.txt`.

Étape BONUS : Extensions Avancées

Maintenant que vous avez une bibliothèque de base, voyons comment vous pouvez l'améliorer encore plus ! Voici quelques idées BONUS pour étendre les fonctionnalités de votre script :

1. Classement des Livres :

- Ajoutez une option pour classer les livres par titre ou par auteur.
- Par exemple, `./bibliotheque.sh classer titre` triera et affichera tous les livres par ordre alphabétique de titre.

2. Notation des Livres :

- Permettez à l'utilisateur d'ajouter une note (de 1 à 5) lorsqu'il ajoute un livre.
- Affichez la note du livre lors de la liste ou de la recherche.

3. Filtrage Avancé :

- Ajoutez des options pour filtrer la liste de livres par note ou par auteur.
- Par exemple, `./bibliotheque.sh lister --auteur="J.K. Rowling"` afficherait tous les livres de cet auteur.

4. Historique des Actions :

- Chaque fois qu'une action est effectuée (ajout, suppression, etc.), enregistrez-la dans un fichier `historique.txt`.
- Ajoutez une option pour afficher l'historique des actions.

5. Backup Automatique :

- Au lieu d'une commande manuelle pour sauvegarder, créez une fonction qui sauvegarde automatiquement la bibliothèque à intervalles réguliers ou après un certain nombre d'actions.

6. Interface Interactive Améliorée :

- Utilisez des outils comme `dialog` ou `whiptail` pour créer une interface utilisateur en mode texte pour votre script, rendant la navigation et la gestion encore plus faciles.

7. Gestion des Erreurs Améliorée :

- Ajoutez plus de vérifications pour les erreurs, comme vérifier si un livre existe déjà avant de l'ajouter ou si un livre n'existe pas avant de le supprimer.

8. Statistiques de la Bibliothèque :

- Créez une option qui affiche des statistiques sur votre bibliothèque, comme le nombre total de livres, l'auteur le plus courant, la note moyenne des livres, etc.

Conseil BONUS : Avant d'implémenter une fonctionnalité bonus, réfléchissez bien à la manière dont elle s'intégrera dans votre script existant. Il peut être utile de dessiner un plan ou un pseudocode pour chaque extension avant de commencer à coder. Bonne chance !

Conseils généraux :

- N'oubliez pas de rendre votre script exécutable avec la commande ``chmod +x bibliotheque.sh``.
- Utilisez des fonctions pour organiser votre code.
- Testez chaque étape avant de passer à la suivante.

Rendu :

À la fin du TP, chaque participant devrait avoir un script fonctionnel qui permet de gérer une bibliothèque. Les participants peuvent être encouragés à ajouter des fonctionnalités supplémentaires ou à améliorer l'interface utilisateur pour rendre le script encore plus robuste.