TP Shell

Exercices bonus

```
$ ./aller_plus_loin.sh --niveau 1 --niveau 2
```

Vous avez terminé le TP ? Ces exercices prolongent vos scripts de création et de suppression des répertoires clients. Traitez les niveaux dans l'ordre ; le défi « Trouve la faille » peut se faire dès le niveau 1 terminé.

Consignes

Travaillez **uniquement sur votre machine virtuelle** : certains scripts suppriment des fichiers.

Gardez une copie de vos scripts du TP avant de les modifier (`cp creer.sh creer_v1.sh`).

Pour chaque exercice, ajoutez à votre compte rendu le script, une copie d'écran de son exécution et une phrase d'observation.

Outil conseillé : `shellcheck` , qui analyse un script et signale les erreurs courantes (Dans le terminal administrateur : `apt install shellcheck` , puis `shellcheck monscript.sh`).

Niveau 1 : rendre le script robuste

Un script utilisé par d'autres doit résister aux erreurs de saisie. Améliorez votre script de création.

1. **Fichier en paramètre.** Le nom du fichier d'utilisateurs est passé en argument : `./creer.sh utilisateurs.txt` .
   - Sans argument, le script affiche `Usage : ./creer.sh fichier.txt` et s'arrête (`exit 1`).
   - Si le fichier n'existe pas, il affiche une erreur et s'arrête (`exit 2`). **Indice :** `[ -f "$1" ]` .
2. **Lignes ignorées.** Les lignes vides et celles qui commencent par `#` ne créent rien.
3. **Pas d'écrasement.** Si le répertoire d'un client existe déjà, le script affiche `[EXISTE] nom` et passe au suivant.
4. **Bilan.** À la fin, le script affiche par exemple `Bilan : 5 créé(s), 2 ignoré(s)` . **Indice :** `compteur=$((compteur + 1))` .
5. **Codes de retour.** Après chaque exécution, tapez `echo $?` . Notez la valeur obtenue dans chaque cas et expliquez à quoi elle sert.

Niveau 2 : sécuriser le script de suppression

Un script qui supprime des fichiers à partir d'un fichier texte est dangereux : celui qui écrit le fichier décide de ce qui sera effacé. Améliorez votre script de suppression.

1. **Valider les noms.** N'acceptez que les noms composés de minuscules, chiffres, `_` et `-`, commençant par une lettre. **Indice :** `[[ "$user" =~ ^[a-z][a-z0-9_-]{1,31}$ ]]`. Tout autre nom est refusé avec un message.
2. **Guillemets partout.** Écrivez `"$BASE/$user"` et non `$BASE/$user`. Testez les deux versions avec un utilisateur nommé `jean dupont` et expliquez la différence.
3. **Mode simulation.** L'option `--dry-run` affiche ce qui serait supprimé, sans rien supprimer.
4. **Confirmation.** Avant chaque suppression, le script demande `Supprimer alice ? (o/N)`. **Attention :** dans une boucle `while read ... done < fichier`, un `read` lit le fichier, pas le clavier. Cherchez comment lire le clavier (`/dev/tty`).
5. **Archiver avant de supprimer.** Chaque répertoire est d'abord sauvegardé dans `~/archives_clients/nom_date.tar.gz` (commande `tar -czf`).
6. **Journal.** Chaque action (création, suppression, refus) est ajoutée à `~/gestion_clients.log` avec la date : `echo "$(date '+%F %T') suppression $user" >> fichier.log`. Seul le propriétaire peut lire ce journal.

À vous de réfléchir

Question : pourquoi est-il préférable de garder une archive plutôt que de supprimer directement ? Pensez à une erreur de manipulation, puis aux obligations d'un hébergeur envers les données de ses clients.