

Mapping certification LPI Linux Essentials - Automatisez vos actions récurrentes avec des scripts Bash

Cette correspondance vous sera surtout utile si vous envisagez un jour de préparer la certification LPI Linux Essentials. Elle vous permet de situer ce cours dans le référentiel, mais ne remplace pas à elle seule une préparation complète à l'examen.

Thème 2 – Utiliser la ligne de commande

- Créer et éditer un fichier de script Bash depuis le terminal.
- Utiliser un éditeur en ligne de commande pour rédiger un script.
- Exécuter un script depuis le dossier courant.
- Utiliser des commandes simples dans un script comme `echo`, `cat` et `read`.
- Passer des arguments à un script avec `$1`, `$2` et `$@`.
- Utiliser des variables pour stocker et réutiliser une information dans un script.

Thème 3 – La puissance de la ligne de commande

- Comprendre ce qu'est un **script Bash** et dans quels cas l'utiliser pour automatiser des tâches répétitives.
- Ajouter un **shebang** (`#!/bin/bash`) pour indiquer l'interpréteur à utiliser.
- Rendre un script exécutable avec `chmod +x`.
- Construire des scripts réutilisables grâce aux **variables** et aux **arguments**.
- Interagir avec l'utilisateur via `read -p`.
- Contrôler l'exécution d'un script avec des **conditions** `if / then / else`.
- Tester l'existence et la nature d'un fichier avec les opérateurs `-e` et `-f`.
- Répéter un traitement sur plusieurs éléments avec une **boucle** `for / in / do / done`.
- Exploiter les **codes de retour** avec `$?` et arrêter proprement un script avec `exit`.
- Sécuriser globalement l'exécution avec `set -e`.

- Intégrer dans un script des chaînes de commandes utilisant les **pipes** (`|`).
- Structurer une commande longue sur plusieurs lignes avec l'antislash `\`.
- Automatiser des analyses avec `grep`, `sort`, `uniq -c`, `wc -l`, `awk` et `head`.
- Produire automatiquement une sortie synthétique lisible à partir de données textuelles ou structurées.
- Planifier l'exécution d'un script avec **cron** et `crontab -e`.

Thème 4 – Le système d'exploitation Linux

- Comprendre qu'un script Bash est un **fichier texte** contenant une suite de commandes exécutées par le système.
- Manipuler des fichiers d'analyse, des logs et des archives dans un environnement Linux.
- Utiliser des outils natifs de Linux pour automatiser des tâches courantes d'exploitation : analyse de fichiers, transformation de données, sauvegarde.
- Découvrir le rôle d'un planificateur système comme `cron` dans l'automatisation des tâches récurrentes.

Thème 5 – Sécurité et bonnes pratiques

- Comprendre qu'un fichier nouvellement créé ou téléchargé n'est pas exécutable par défaut pour des raisons de sécurité.
- Ajouter manuellement les droits d'exécution uniquement lorsque c'est nécessaire.
- Vérifier l'environnement avant d'agir pour éviter les erreurs en cascade.
- Afficher des messages explicites pour guider l'utilisateur en cas d'erreur.
- Gérer les erreurs proprement avec `$?` , `exit` et `set -e`.
- Journaliser l'exécution d'un script grâce aux redirections `>>` et `2>`.
- Vérifier qu'une sauvegarde s'est bien déroulée et conserver une trace exploitable dans un fichier log.
- Rendre un script de sauvegarde **relançable** et éviter l'écrasement d'archives grâce à un horodatage.
- Comprendre les principes de base d'une sauvegarde fiable : nommage clair, chemin maîtrisé, gestion des erreurs, possibilité de restauration.
- Sensibiliser à la nécessité d'**externaliser** une vraie sauvegarde hors du disque source.

À noter

- Ce cours couvre **très directement** la partie du référentiel liée au **shell scripting de base** : création de scripts, permissions d'exécution, variables, arguments, conditions, boucles et gestion simple des erreurs.
- Il prolonge aussi des compétences de ligne de commande en les mobilisant dans un **cadre automatisé, structuré et reproductible**.
- Il touche également à des notions d'exploitation système importantes, notamment la **journalisation**, la **sauvegarde**, la **planification** et les premiers réflexes de robustesse d'un script.
- En revanche, il ne vise pas un niveau avancé de développement logiciel : l'objectif reste l'automatisation pratique de tâches Linux courantes.