

# TD6 – Langages de script

Abdallah Ammar

12 janvier 2026

## Objectifs du TD

Ce TD a pour objectif de mobiliser l'ensemble des notions vues précédemment afin de réaliser un script bash complet et utile. À l'issue de la séance, vous devrez être capables de :

- analyser une consigne complète;
- écrire un script bash structuré;
- automatiser des traitements sur des fichiers texte;
- gérer des erreurs simples;
- produire une sortie lisible.

Ce TD est un mini-projet. Prenez le temps de lire attentivement la consigne avant d'écrire votre script.

## 1 Contexte

Vous disposez de fichiers texte contenant des informations sur des utilisateurs et des événements (logs). Ces fichiers sont ceux utilisés lors du TD3.

L'objectif est d'écrire un script bash unique permettant d'analyser automatiquement ces fichiers.

## 2 Consignes générales

- Le script doit s'appeler `analyze.sh`.
- Il doit être exécutable.
- Il prend un répertoire en paramètre.
- Il doit vérifier que ce répertoire existe.
- Il doit produire une sortie lisible et structurée.
- L'utilisation d'éditeurs graphiques est interdite.

## 3 Étape 1 – Vérification des paramètres

### Travail à faire

1. Si aucun paramètre n'est fourni, affichez un message d'erreur.
2. Si le paramètre fourni n'est pas un répertoire, affichez un message d'erreur.

## Indications

Vous pouvez utiliser :

- les paramètres du script (**\$1**) ;
- le test `-d`.

## 4 Étape 2 – Analyse des utilisateurs

À partir du fichier `users.txt` présent dans le répertoire fourni :

### Travail à faire

1. Affichez le nombre total d'utilisateurs.
2. Affichez la liste des shells utilisés.
3. Indiquez le shell le plus utilisé.

## Indications

Vous pouvez réutiliser les commandes vues en TD3 et TD5 :

- `cut`
- `sort`
- `uniq`
- `wc`

## 5 Étape 3 – Analyse des logs

À partir du fichier `logs.txt` présent dans le répertoire fourni :

### Travail à faire

1. Affichez le nombre total d'événements.
2. Affichez le nombre d'erreurs.
3. Affichez la liste des utilisateurs ayant généré une erreur.

## Indications

Vous pouvez utiliser :

- `grep`
- `cut`
- `sort`
- `uniq`

## 6 Étape 4 – Affichage structuré

Votre script doit afficher les résultats de manière claire, avec des titres explicites pour chaque partie de l'analyse.

## Exemple indicatif de sortie

```
=== Analyse des utilisateurs === Nombre total d'utilisateurs : 5 Shells utilisés :  
/bin/bash (3) /bin/zsh (1) /usr/sbin/nologin (1)  
=== Analyse des logs === Nombre total d'événements : 20 Nombre d'erreurs : 4  
Utilisateurs ayant généré une erreur : alice bob
```

Cet exemple est fourni à titre indicatif. La présentation exacte peut varier.

## 7 Extensions (optionnelles)

Pour les étudiants les plus rapides :

- vérifier la présence des fichiers avant analyse ;
- sauvegarder les résultats dans un fichier ;
- ajouter une option `-h` affichant une aide ;
- analyser automatiquement tous les fichiers `.txt` du répertoire.

## Remarques

- Ce TD ne demande aucun concept nouveau.
- Réutilisez les commandes et scripts déjà écrits précédemment.
- Testez votre script avec différents cas.
- Un script clair et simple est préférable à un script complexe.

## Pour aller plus loin – Exercice ludique (optionnel)

Recherchez dans les fichiers de logs :

- les messages contenant `WARNING` ;
- les messages contenant `ERROR` ;
- les messages contenant votre prénom (s'il y en a).

Expliquez ce que vous observez.