

TD10 – Langages de script

Abdallah Ammar

12 janvier 2026

Objectifs du TD

Ce TD a pour objectif d'introduire la gestion des sorties et des erreurs sous Linux. À l'issue de la séance, vous devrez être capables de :

- comprendre la différence entre sortie standard et erreur standard ;
- rediriger les sorties vers des fichiers ;
- analyser des messages d'erreur ;
- améliorer la robustesse de scripts bash simples.

Ce TD introduit de nouveaux concepts. Il n'est pas attendu que tout soit parfaitement maîtrisé : l'objectif est de comprendre les idées générales.

1 Les flux standards

Sous Linux, un programme communique avec l'extérieur à l'aide de flux :

- l'entrée standard (stdin) ;
- la sortie standard (stdout) ;
- la sortie d'erreur (stderr).

Par défaut, la sortie standard et la sortie d'erreur sont affichées à l'écran.

2 Rediriger la sortie standard

Travail à faire

1. Listez le contenu de votre répertoire personnel.
2. Redirigez cette sortie dans un fichier `liste.txt`.
3. Vérifiez le contenu du fichier.

Commandes

```
$ ls $ ls > liste.txt $ cat liste.txt
```

3 Générer et observer une erreur

Travail à faire

1. Essayez d'afficher un fichier qui n'existe pas.
2. Observez le message affiché.

```
$ cat fichier_inexistant.txt
```

Expliquez pourquoi le message n'est pas redirigé avec >.

4 Rediriger la sortie d'erreur

La sortie d'erreur peut être redirigée séparément.

Travail à faire

1. Redirigez l'erreur précédente dans un fichier `erreur.txt`.
2. Vérifiez le contenu du fichier.

```
$ cat fichier_inexistant.txt 2> erreur.txt $ cat erreur.txt
```

5 Rediriger sortie et erreur

Il est possible de rediriger à la fois la sortie standard et la sortie d'erreur.

Travail à faire

1. Lancez une commande produisant à la fois une sortie et une erreur.
2. Redirigez les deux flux dans un même fichier.

```
$ ls fichier_inexistant > sortie.txt 2> erreur.txt
```

Ou en une seule commande :

```
$ ls fichier_inexistant > tout.txt 2>&1
```

6 Utilisation dans un script

Les redirections sont très utiles dans les scripts pour éviter d'afficher des messages inutiles à l'utilisateur.

Travail à faire

1. Reprenez un script écrit dans les TD précédents.
2. Redirigez les messages d'erreur vers un fichier `error.log`.
3. Vérifiez que le script continue de fonctionner.

7 Rendre un script plus robuste

Travail à faire

Écrivez un script qui :

- prend un fichier en paramètre ;
- vérifie qu'il existe ;
- affiche son contenu s'il existe ;
- affiche un message d'erreur clair sinon.

Comparez le comportement du script avec et sans redirections.

Ouverture

La gestion des erreurs est essentielle dans les scripts réels. Dans des modules plus avancés, vous verrez :

- les codes de retour ;
- la gestion des signaux ;
- les logs système ;
- les scripts robustes à grande échelle.