

TD0 – Langages de Script

Abdallah Ammar

19 janvier 2026

Ce TD a pour objectif de préparer l'environnement de travail pour le module **Langages de script**. Les outils et scripts utilisés dans ce cours s'exécuteront dans un environnement **Linux / Unix**.

Le symbole \$ affiché au début des commandes correspond à l'invite du terminal. Il sert uniquement à indiquer que la commande doit être tapée, mais ne fait pas partie de la commande elle-même.

- **Étudiants sous Windows** : ce TD vous guide pas à pas dans l'installation d'un environnement Linux grâce à **WSL (Windows Subsystem for Linux)**.
- **Étudiants disposant déjà de Linux** : aucune installation n'est requise. Vous pouvez passer directement aux prochains TD.
- **Étudiants sous macOS** : aucun système supplémentaire n'est nécessaire. Le terminal est disponible par défaut et le shell **bash** est déjà installé. Depuis les versions récentes de macOS, le shell utilisé par défaut est **zsh**. Pour connaître le shell actuellement utilisé, tapez la commande :

```
$ echo $SHELL
```

Si nécessaire, il est possible de lancer **bash** simplement en tapant :

```
$ bash  
$ # ou  
$ /bin/bash
```

dans le terminal. Pour confirmer que **bash** est actif, utilisez cette commande :

```
$ echo $0
```

Optionnellement, pour définir **bash** comme environnement par défaut, utilisez la commande :

```
$ chsh -s /bin/bash
```

Pour installer des programmes supplémentaires en ligne de commande, il est recommandé d'utiliser le gestionnaire de paquets **Homebrew**. L'installation de Homebrew se fait une seule fois avec les commandes suivantes :

```
$ brew_link="https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh"  
$ curl -fsSL $brew_link -o install_brew.sh  
$ bash install_brew.sh
```

Après l'installation, si la commande **brew** n'est pas reconnue,

```
$ brew --version
```

il est nécessaire d'ajouter Homebrew au PATH.

```
$ echo 'eval "$( /opt/homebrew/bin/brew shellenv )"' > ~/.bashrc
$ eval "$( /opt/homebrew/bin/brew shellenv )"
```

Qu'est-ce que WSL ?

WSL est une fonctionnalité de Windows qui permet d'exécuter un système Linux directement sous Windows, sans machine virtuelle lourde ni redémarrage.

Avantages :

- Pas besoin de dual-boot
- Intégration avec Windows
- Accès aux commandes Linux standards

Prérequis

- Windows 10 (version 2004 ou plus récente) ou Windows 11
- Connexion Internet
- Droits administrateur sur la machine

Étape 1 : Ouvrir un terminal Windows

1. Appuyez sur Win + R
2. Tapez cmd
3. Appuyez sur Ctrl + Shift + Entrée pour ouvrir en mode administrateur

Étape 2 : Installer WSL

Dans le terminal, tapez la commande suivante :

```
$ wsl --install
```

Cette commande :

- Active WSL
 - Installe le noyau Linux
 - Installe par défaut Ubuntu
- Redémarrez votre ordinateur** lorsque cela vous est demandé.

Étape 3 : Premier lancement d'Ubuntu

Après le redémarrage :

- Lancez **Ubuntu** depuis le menu Démarrer
- Patientez pendant l'installation
- Choisissez un nom d'utilisateur et un mot de passe Linux

Étape 4 : Vérification de l'installation

Dans le terminal Ubuntu, tapez :

```
$ lsb_release -a
```

Puis :

```
$ uname -a
```

Si ces commandes s'exécutent sans erreur, l'installation est réussie.

Problèmes fréquents

- **Erreur 0x80370102** : la virtualisation du CPU n'est pas activée. Il faut activer la virtualisation dans le BIOS/UEFI. Depuis Windows, allez dans *Paramètres* → *Système* → *Récupération* → *Démarrage avancé*. Dans le menu de démarrage, accédez aux options UEFI et activez la virtualisation du CPU (les menus peuvent varier selon les machines). Voir cette discussion par exemple.
- **Erreur 0x800701bc** : WSL est en version 1. Mettez à jour votre système Windows et assurez-vous d'utiliser WSL 2. Voir ici.
- **Erreur 0xc03a001a** : la distribution Ubuntu est installée dans un dossier compressé. Il faut désactiver la compression du dossier `C:\Users\<nom>\AppData\Local\Packages\CanonicalGroupLimited.Ubuntu` et de tous ses sous-dossiers. Si le dossier `AppData` n'est pas visible, activez l'affichage des dossiers cachés. Voir ici.
- **Ubuntu ne se configure pas au premier lancement** : la distribution n'est peut-être pas installée. Installez manuellement Ubuntu 22.04 LTS depuis le Microsoft Store.
- **Erreur 0x80070050** : essayez de mettre à jour Windows ou de réinstaller Ubuntu 22.04. Si le problème persiste, un diagnostic de Windows Update peut être nécessaire.
- **Erreur 0x80004002** : vérifiez que les fonctionnalités de virtualisation (Hyper-V ou *Virtual Machine Platform*) sont activées dans *Fonctionnalités Windows*.
- **Erreur Le chemin d'accès est introuvable** : lors du lancement d'Ubuntu, essayez de mettre à jour WSL en lançant la commande suivante dans le terminal Windows (PowerShell) :

```
$ wsl --update
```