

TD #1 – Manipulation du terminal avec un soupçon de script

Préambule

Objectif : Découvrir les bases du terminal et du langage Shell à l'aide d'exercices simples

Fichiers additionnels :

- `example-script.sh` – script d'exemple sur les structures de contrôle

1) Compréhension des commandes de base

Soit la liste de commandes/exécutables suivante : `cat`, `cd`, `cp`, `diff`, `echo`, `python`, `ls`, `man`, `mkdir`, `mv`, `rm`, `rmdir`, `sudo`, `tar`, `time`, `touch`, `vi`.

1. Donnez une brève description pour chacune des commandes.
2. Quelles commandes consistent en l'exécution d'un binaire ?
3. Quels chemins sont représentés par les symboles `.`, `..` et `~` ?

2) Première utilisation du terminal

Pour chacune des questions suivantes, exécutez la commande correspondante.

1. Déplacez-vous dans le répertoire temporaire de votre système de fichiers.
2. Créez le répertoire `project`, ainsi que les sous-répertoires `data`, `doc` et `module`.
3. Au sein du répertoire `project`, créez un fichier `README` contenant votre nom et votre prénom. Créez le fichier `matrix.csv` dans le répertoire `data` et les fichiers `main.py` et `core.py` dans le répertoire `module`.
4. Affichez la hiérarchie complète du répertoire `project` et de ses sous-répertoires, puis écrivez ce résultat dans `contents.txt`.
5. Créez une copie du répertoire `project` que vous nommerez `projectV2`. Supprimez le répertoire `project`.
6. Créez l'archive `pv2.tar` contenant l'ensemble du répertoire `projectV2`.

3) Premier script Shell

Afin d'automatiser l'exécution de commandes, comme par exemple la compilation d'un projet ou l'exécution d'un jeu de tests, il est possible de les rassembler dans un fichier. Ce type de fichier est appelé script.

Placez l'ensemble des commandes écrites dans l'exercice précédent dans un script Shell, et exécutez le. Le résultat est-il le même que précédemment ?

4) Shell en Python

Il vous est possible d'exécuter des commandes Shell en Python, soit en exécutant un script à l'aide (entre autres) de la fonction `os.exec1()`, soit en exécutant une commande avec l'appel `os.system()`.

À l'aide de cette dernière, faites un programme python qui affiche le contenu de votre répertoire personnel.

5) Est-ce que scripter c'est développer ? [optionnel]

Un script Shell peut, tout comme un programme Python, prendre des arguments lors de son exécution. Supposons la commande `./script.sh hello world` : les arguments `$0`, `$1` et `$2` contiennent respectivement `./script.sh`, `hello` et `world`.

Un script Shell peut également utiliser des blocs de contrôle (condition, boucle, etc.). Le script `shell-example.sh` vous montre quelques exemples d'utilisation de ces blocs.

Somme des entiers

Écrivez un script Shell qui prend un entier positif n en paramètre et affiche la somme des entiers de 0 jusqu'à n .

Calculatrice

Écrivez un script Shell qui prend trois arguments `a`, `b`, et `op`, avec `a` et `b` des entiers et `op` une opération parmi `{+, -, *, +}` et affiche le résultat de `a op b`.