

# ISS - Initiation aux Systèmes d'exploitation et au Shell

## LU2IN020

### TP 01 – Introduction au Bash

Julien Sopena

septembre 2022

Le but de cette première semaine est de comprendre le fonctionnement de l'interpréteur de commandes bash et d'apprendre à réaliser un petit script.

#### Exercice 1 : Balade le long des chemins

Pour se situer dans l'arborescence, on peut utiliser la commande `pwd` ou afficher le contenu de la variable `$PWD`.

Pour vous déplacer dans l'arborescence des fichiers, vous pouvez utiliser la commande `cd` (pour *change directory*) qui vous positionnera dans le répertoire dont le chemin est passé en paramètre.

Un chemin est une succession de répertoires séparés par `/` (*e.g. rep1/rep2/rep3*), l'origine peut être :

- le répertoire courant par défaut
- `./` : le répertoire courant
- `../` : le répertoire parent du répertoire courant
- `/` : la racine du système, on parle alors de chemin absolu
- `~/` : votre *Home*

#### Question 1

En remontant progressivement dans l'arborescence avec des `cd ...`, tentez d'atteindre la racine du système. Que se passe-t-il lorsqu'on essaye de remonter plus haut ?

Pour éviter de retaper vos commandes, vous pouvez utiliser l'historique de votre Shell avec la touche "flèche vers le haut".

#### Question 2

Que se passe-t-il si vous exécutez la commande `cd` sans paramètre ?

#### Question 3

Retournez à la racine en une seule commande.

#### Question 4

Revenez enfin dans votre *home* par une succession de commandes `cd` en utilisant au maximum la touche *tabulation* qui complète votre frappe dès qu'il n'y a plus d'ambiguïté.

## Question 5

Comparez maintenant le résultat des trois commandes suivantes. Quelle conclusion pouvez-vous faire sur le mécanisme de substitution du *Bash* ?

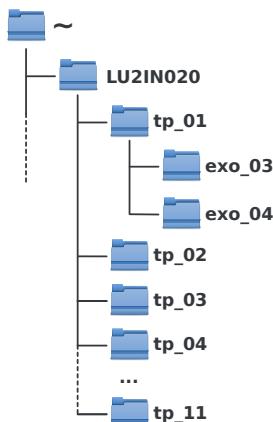
1. `moi@pc ~ $ ls ./toto/titi`

2. `moi@pc ~ $ ls ~/toto/titi`

3. `moi@pc ~ $ ls ../toto/titi`

## Exercice 2 : Création de votre arborescence de travail

Comme il devient rapidement impossible de travailler directement dans sa **Home**, nous vous demandons de travailler ce semestre en organisant vos fichiers sous la forme de l'arborescence suivante :



## Question 1

La commande `mkdir` permet de créer les répertoires dont les chemins sont passés en paramètre. Quel problème pose la commande suivante :

`moi@pc ~ $ mkdir /tmp/rep1/rep2`

Trouvez une solution en utilisant le manuel de la commande accessible en lançant la commande :

`moi@pc ~ $ man mkdir`

## Question 2

Utilisez intelligemment la commande `mkdir` et ce que vous avez appris en TD pour générer l'arborescence demandée. Vous pouvez vérifier le résultat avec la commande `tree` qui affiche l'arborescence des fichiers depuis le répertoire passé en paramètre (répertoire courant par défaut en absence de paramètre).

## Exercice 3 : Jeu de piste historique

### Question 1

Après vous être placé dans le répertoire correspondant au deuxième exercice, vous allez devoir récupérer une archive contenant les données du jeu de piste à l'aide de la commande `wget`. Cette commande du projet *GNU* permet de récupérer du contenu d'un serveur Web ou FTP. Ce contenu peut être une simple page web, mais aussi un document, voir l'ensemble d'un site web. Assurez-vous, que le fichier ait bien été créé dans le répertoire de l'exercice avant de continuer.

```
moi@pc /home/moi $ wget http://julien.sopena.fr/LU2IN020-TP_01-EXO_03.tgz
```

### Question 2

Le fichier que vous venez de charger est une archive `tar`. À la différence d'un fichier `zip`, qui est une version compressée d'un fichier, une archive est une représentation de toute une arborescence (répertoires et fichiers, ainsi que leurs droits) dans un seul fichier. Cela est très utile pour envoyer tout un projet en pièce jointe dans un mail.

L'extension `.tgz`, qui est parfois noté `.tar.gz`, indique que l'archive a été compressée pour en réduire la taille.

Pour extraire les fichiers de l'archive, vous allez utiliser la commande ci-dessous. Mais avant utilisez la commande `man` pour afficher son contenu avant de l'extraire réellement.

```
moi@pc /home/moi $ tar xzf LU2IN020-TP_01-EXO_03.tgz
```

### Question 3

Avant d'aller plus loin et après avoir vérifié que le contenu de l'archive est bien présent dans votre répertoire, effacez cette dernière avec la commande `rm`. Cette commande effacera tous les fichiers et répertoires dont les chemins sont passés en paramètre (dans le cas de répertoires, il faudra ajouter l'option `-r`).

### Question 4

Vous allez maintenant pouvoir vous lancer dans le jeu de piste. En parcourant les fichiers, d'indice en indice, trouvez une phrase qui montre la nécessité d'avoir un bon interprète. Et comme souvent tout commence par un `README`.

## Exercice 4 : Du son en boucle

### Question 1

Pour commencer, placez le contenu de l'archive `http://julien.sopena.fr/LU2IN020-TP_01-EXO_04.tgz` dans le répertoire `exo_04` de votre arborescence. Pour économiser de la place, n'oubliez pas d'effacer l'archive après avoir récupéré son contenu.

### Question 2

Avant d'aller plus loin, vous allez devoir nettoyer le répertoire extrait en supprimant l'ensemble des fichiers `log`. Il ne devrait plus rester que des fichiers `flac` (codec libre de compression audio sans perte)



Nous le verrons plus tard, mais pour des raisons de droits vous devez confirmer la suppression des fichiers. Vous allez vite vous rendre compte que de valider à la main chaque suppression est dans ce cas très pénible.

Heureusement, la plupart des commandes s'accompagne d'options permettant de modifier le comportement de la commande. Les options se déclarent avec un tiret suivi d'une lettre (ou de deux tirets suivis d'un mot pour les options dites longues). Il est recommandé par la norme de les placer avant les autres paramètres.

Dans notre cas, vous pouvez forcer la suppression avec l'option `-f` (ou `--force`). **ATTENTION : l'utilisation de l'option -f de la commande rm est dangereuse puisqu'elle conduit à effacer tout ce que l'on demande sans aucune confirmation.**

### Question 3

La base de travail est presque prête pour l'exercice, il vous faut encore effacer les deux fichiers suivants : `error.flac` et `*.flac`

### Question 4

Il ne devrait maintenant rester dans le répertoire qu'un ensemble de fichiers `flac` correspondant à autant d'albums de musique. Le nom de chacun de ces fichiers suit le format suivant : `<Artiste>-<Titre>-<Année>`.

On veut maintenant afficher tout un ensemble d'informations sur ces 1001 albums. Réaliser ce travail directement au sein de la console pourrait s'avérer fastidieux. Il devient alors nécessaire de développer un script, *i.e.*, une succession de commandes rassemblées au sein d'un fichier.

Vous allez donc ajouter toutes les commandes des prochaines questions dans un fichier nommé `infos_1001_album.sh` que vous placerez dans le répertoire `exo_04`. Après vous être placé dans ce répertoire, vous pourrez alors lancer le script `infos_1001_album.sh` avec la commande suivante :

```
moi@pc /home/moi $ bash infos_1001_album.sh
```

Pour commencer, on veut afficher la phrase "**Les albums recommandés des Beatles sont :**" suivi de la liste de tous les albums des 4 de Liverpool présents dans la sélection.

### Question 5

Ajoutez maintenant, à votre script l'affichage du ou des albums dont le nom est une simple lettre.

### Question 6

On veut maintenant afficher le nombre d'albums réellement présents dans cette sélection. Pour ce faire, vous allez devoir implémenter un compteur en utilisant la substitution arithmétique `$(( expression ))`. Toutes les opérations arithmétiques simples (+, -, \*, /, %) sont disponibles et l'expression peut comporter des variables de votre Script.

### Question 7

Ajoutez ensuite à votre script le calcul et l'affichage de la moyenne des tailles des noms de fichiers de ces albums.

### Question 8

Afin d'avoir une représentation graphique de la répartition de ces albums suivant les années (de 1995 à 2016), ajoutez à votre script les instructions qui réaliseront l'affichage suivant où chaque '**x**' représente un album :



```
1955 : x
1956 : xxxxxx
1957 : xxxxxx
1958 : xxxxx
...
```

### Question 9

Pour finir, utilisez la commande `cp` permettant de copier un ou plusieurs fichiers (voir son manuel) pour implémenter un nouveau script `sort_album_by_year.sh` qui :

- crée dans un répertoire `by_year` un répertoire pour toutes les années de 1955 à 2016 ;
- copie chaque album dans le répertoire correspondant à son année.

