

TD principe et fonctionnement systeme info

nicolas.carrara1u

September 2018

Tout les exercices se font via le terminal. Normalement, aucun cliquer de souris ne devrait être requis.

1 Exercice 1

1. Allez sous votre home.
2. Créez un dossier TD1. Quel est son chemin absolu ?
3. Créez le dossier TD1/exercice1. Allez sous ce dossier.
4. Créez un fichier vide avec la commande `touch`.
5. Renommer ce fichier en `test.sh`.
6. Lancez la commande `vim test.sh`. Vim est un éditeur de texte en mode terminal.
7. Pour passer en mode édition, cliquez sur la touche **inser**.
8. Ecrivez l'entête des fichiers sh.
9. Sauter une ligne. Ecrivez `echo "mon premier fichier linux"`.
10. Ouvrez un autre terminal (`ctrl+alt+t` ou `ctrl+maj+t`). Tapez la commande `man echo`. Que fait la commande `man` ? Et la commande `echo` ?
11. Retourner sur le terminal de départ. Vim devrait être toujours ouvert.
12. Sauvegardez votre fichier. Pour cela, passez en mode commande line. Cliquez sur la touche **echap** puis `:`. Tapez `w` (pour write) puis **entrée** pour sauvegarder le fichier.
13. Toujours en mode commande line, quittez le fichier avec la commande `q` (pour quit).
14. Si vous l'exécutez, que va faire le fichier `test.sh` ?
15. Exécutez ce fichier avec la commande `sh` ou `./`.
16. Que s'est il passé, et pourquoi ?
17. Vérifier les droits en execution du fichier.
18. Rendez le fichier executable, puis exécutez le.
19. Nettoyez le terminal avec la commande `clear`.
20. Fermez ce terminal avec la commande `exit`.

2 Exercice 2

1. Ouvrez un terminal.
2. Créer le dossier TD1/exercice2. Allez sous ce dossier.
3. Créer un fichier python qui fait une boucle infinie. Toute les 100 itérations, le programme doit afficher l'itération courante.
4. Lancez le fichier python.
5. Tuez le processus python avec le raccourci "ctrl+c".
6. Relancez le programme python.
7. Ouvrez un autre terminal. Tapez la commande `top`. Que remarquez vous ?
8. Tuez le processus python en utilisant l'interface de la commande `top` (chercher comment faire ...).
9. Relancer le programme python.
10. Ouvrez un terminal et lancez une deuxième fois le programme. Vous devriez avoir deux programmes python qui tournent en parallèle.
11. Retournez dans `top`. Que remarquez vous ?
12. Quittez l'interface `top`. Sur ce même terminal lancez `pkill -9 "python"` pour tuer tous les processus python en cours. Utiliser cette commande que si vous savez quels processus avec le nom python tournent actuellement.
13. Fermez tous les processus et tous les terminaux en cours.

3 Exercice 3

1. Récupérez l'archive de fichiers multimédia : http://ncarrara.fr/others/dataset_unsorted.tar.gz
2. Créez le dossier TD1/exercice3.
3. Déplacer l'archive dans ce dossier.
4. Copier une version de l'archive dans TD1.
5. Extraire tous les fichiers de l'archive avec la commande `tar` (utilisez `man` pour comprendre comment utiliser la commande).
6. Créez un programme `sh` qui créer une arborescence de ces fichiers en fonction du type (musique, vidéo, image ...), de la saison (pour les séries), de l'auteur et de l'album (pour les musiques). Mettre tous les films dans le même dossier. Voici ce que cela doit donner <http://ncarrara.fr/others/tree.png>
7. Utilisez la commande `tree` pour vérifier que l'arborescence est bonne.

4 Exercice 4

Faire tous les exercices du cours 2. Les tester sur machine. Pour les regex, utiliser python (et sa syntaxe associée : <https://docs.python.org/3/library/re.html>). Attention, la syntaxe peut différer de la norme posix BRE ou ERE. Tester vos regex avec des exemples. Correction au prochaine TD.

5 Exercice 5

Renommer (avec la commande linux rename) tous les fichiers multimédias de l'exercice 3 sous les formes suivantes :

— Les musiques :

```
artiste="linkin-park"_album="meteora"_songname="Foreworld"_id="01".mp3
```

— Les séries :

```
tvshow="Rick and Morty"_saison="01"_episode="01"_episodename="Pilot".[mkv|avi]
```

— Les films :

```
movie="starwars_1".[mkv|avi]
```

6 Projet : gestionnaire de bibliothèque multimédia

Lire l'énoncé entièrement avant de commencer la programmation. Barème à titre indicatif.

Objectif Créer un utilitaire linux en python pour la gestion de fichiers multimédia. La bibliothèque à gérer est celle de l'exercice 3.

Pré requis

- Trouver un binome (je ne veux pas de groupe de 3). Vous aurez une note commune.
- Les fichiers doivent être triés et renommés (exercice 3 et 5).
- Télécharger VLC pour linux.
- Comprendre comment lancer VLC via la ligne de commande.

Fonctionnalités à implémenter

- Afficher l'aide (0.5).
- Afficher la bibliothèque complète (comme le résultat de la commande Tree) (0.5).
- Afficher tous les films/musiques/series (1).
- Afficher toutes les musiques d'un album, d'un artiste (trier selon les albums).
- Lancer (avec VLC) un fichier en particulier (2).

- Ajouter un fichier à la bibliothèque, en précisant les métadonnées (id, saison, episode, name, album) (utiliser le prompt `https://anh.cs.luc.edu/python/hands-on/3.1/handsonHtml/io.html`) selon le type de fichier (film/musique/tvshow) et le path du fichier à ajouter. Renommer le fichier en conséquence et l'insérer dans le bon dossier de la bibliothèque (créer de nouveau dossier si l'auteur n'existe pas, par exemple) (4).
- Supprimer des fichiers grâce à une regex (par exemple "RickAndMortyS01E++") (2)
- Afficher des messages d'erreurs quand les inputs ne sont pas respectés (ou autres) (2)

Important

- Les conventions à respecter :
`http://pubs.opengroup.org/onlinepubs/9699919799/basedefs/V1_chap12.html`
- Utiliser la bibliothèque `argparse` pour les options, arguments etc :
`https://docs.python.org/3.3/library/argparse.html`
- Vous êtes libre d'utiliser les fonctions python pour gérer les fichiers (`https://docs.python.org/3/library/os.html`) OU de lancer des commandes linux (`mv`, `rm`, `cp` ...) via python (`https://docs.python.org/3/library/subprocess.html`).

Notations

Vous serez notés selon :

- Les fonctionnalités implémentées. (12)
- La robustesse des fonctionnalités. Imaginez qu'un utilisateur fasse n'importe quoi avec l'utilitaire. Vous devez prendre en compte tous les cas possible. (3)
- Le respect des conventions. (3)
- L'ergonomie de l'utilitaire. (2)

Vous présenterez votre utilitaire pendant une dizaine de minutes. Pour vous aider, créer un fichier SH avec toutes les commandes que vous lancerez pendant la présentation (fichier à rendre). Pendant 5 minutes supplémentaires, je mettrai votre utilitaire à l'épreuve et vous poserai des questions.