

TD #4 – Répertoires et méta-données

Préambule

Objectif : Reproduction du comportement de commandes shell liées à la manipulation des répertoires et des méta-données de fichier

Fichiers additionnels :

- `dircore.py` – programme contenant les prototypes de fonctions des énoncés du TD ; ne changez pas les prototypes

Fonctions utilitaires

De l’octal à la chaîne de caractères

Il existe deux représentations possibles d’un mode : (1) l’octal et (2) la chaîne de caractères (cf. cours). Écrivez le corps des fonctions `mode_octal_to_str()` et `mode_str_to_octal()` permettant de passer de l’une des représentations à l’autre.

Modification des méta-données d’un fichier

chmod – Modification des permissions

Écrivez le corps de la fonction `change_mode()` qui permet de modifier les permissions du fichier `path_in` par la valeur octale `new_mode`.

touch – Modification de la date de dernière accès/modifications

Écrivez le corps de la fonction `touch()` qui permet de modifier la date de dernier accès et modification du fichier `path_in` par la date courante, ou de créer le fichier s’il n’existe pas. N’hésitez pas à regarder la fonction `os.utime()`.

Affichage du contenu d'un répertoire

ls – Affichage simple

Écrivez le corps de la fonction `dir_list()` qui permet de lister sur la sortie standard les noms des fichiers et répertoires contenus dans `path_in`.

ls -a – Affichage détaillé

Écrivez le corps de la fonction `dir_all_list()` qui permet de lister sur la sortie standard les fichiers/répertoires ainsi que leurs méta-données associées contenus dans `path_in`. L'affichage doit être formaté de la manière suivante :

```
>>> list.dir_all_list(path_in)
# mode user size last_access_time name
rwx-w-r-- username 412 24 sept. 2021 filename
```

N'hésitez pas à regarder les fonctions `pwd.getpwuid()` et `time.gmtime()`.

ls -R – Affichage récursif

Écrivez le corps de la fonction `dir_rec_list()` qui permet de lister sur la sortie standard les fichiers/répertoires récursivement à partir de `path_in`. L'affichage doit être formaté de la manière suivante :

```
>>> list.dir_rec_list(path_in)
dir_A
dir_A/filename
dir_B
dir_B/dir_b/filename
dir_B/filename
dir_B/filename_2
filename
filename_2
```

Attention : l'ordre d'affichage des entrées n'est PAS important.