

# Laboratoire 2 - Linux

## 1 Durée prévue : 0h30

## 2 Exercice 1 – opération sur les dossiers :

### 2.1 Avant de commencer

Il faut savoir que Linux est sensible à la casse (case sensitive en anglais), c'est à dire qu'il distingue les majuscules des minuscules. Ainsi, pour créer un répertoire, la commande est **mkdir**, ce n'est pas la peine d'essayer **MKDIR** ou **mKdiR**, cela ne fonctionnera pas.

Sous Linux, les chemins sont séparés par des slash : **/etc/init/xfs** mais jamais **\etc\init\xfs**.

### 2.2 Les commandes pour gérer les dossiers et fichiers

#### **cd**

change directory, la commande permet de naviguer dans l'arborescence. Par exemple **cd /var/log** va dans le dossier des logs, quel que soit l'endroit où l'on se trouve puisqu'on a mis le slash de début, lequel indique qu'il s'agit d'un chemin absolue. En revanche, **cd mondossier/images** va dans le répertoire images de mondossier lequel se trouve à l'endroit où on se situe déjà. Comme on ne met pas de slash de début, il s'agit d'un chemin relatif, on ajoute donc ce chemin à celui dans lequel on se trouve déjà. Deux raccourcis très utiles sont à connaître. **cd ~** mène au répertoire de l'utilisateur courant (**/home/user/** la plupart du temps ou **/root/** si vous êtes en root) et **cd -** retourne sur le chemin précédent.

#### **ls**

liste les fichiers d'un dossier. Options : **-a** pour les fichiers cachés, **-l** pour la liste détaillée, **-h** pour les tailles en unités « human readable ». Très pratique l'option **-R** permet en un coup d'oeil de visualiser les sousdossiers.

#### **pwd**

print working directory. Cette commande affiche tout simplement le chemin absolu du dossier dans lequel on se trouve.

#### **touch**

crée un fichier. Il suffit de faire **touch nom\_du\_fichier**.

**touch** a pour but premier de modifier l'horodatage d'un fichier. Si vous faites **touch** sur un fichier qui existe déjà, cela actualisera ses dates de dernier accès et modification.

#### **mkdir**

crée un dossier, le fonctionnement est le même que celui de la commande **touch**.

**mkdir nom\_du\_dossier**

#### **cp**

copy, fait une copie d'un fichier. L'option **-R** permet de réaliser des copies de dossiers entiers.

**cp fichier\_original copie\_du\_fichier**

# on peut aussi placer la copie directement dans un autre dossier

**cp fichier\_original nom\_du\_dossier/copie\_du\_fichier**

**mv**

move, permet de déplacer des fichiers/dossiers. La commande **mv** s'utilise exactement de la même manière que la commande **cp**. En outre, cette commande permet aussi de renommer les fichiers et dossiers en changeant simplement leur nom.

**mv mon\_fichier mon\_fichier\_new\_name**

**rm**

remove, supprime des fichiers. **rm nom\_du\_fichier**. L'option **-f** force la suppression, l'option **-i** demande une confirmation avant suppression, enfin l'option **-r** permet la suppression récursive des dossiers et sous-dossiers.

**rmdir**

remove directory, supprime un dossier seulement s'il est vide.

### 2.3 Répertoires spéciaux :

- . représente le répertoire courant,
- .. représente le répertoire parent
- ~ représente le répertoire maison (home) de l'utilisateur

### 2.4 Fichiers cachés :

Sous Linux, les fichiers cachés commencent par un point. Par exemple, **~/ .bashrc** est un fichier caché, dans le répertoire maison de l'utilisateur, qui contient la configuration de son **shell**.

### 2.5 Jokers : ? et \*

Les caractères **?** et **\*** dans les noms de fichiers et de répertoires permettent de représenter des caractères quelconques. **'?'** représente un seul caractère, tandis que **'\*'** en représente un nombre quelconque.

Par exemple, « **\*.jpg** » représente tous les fichiers se terminant par jpg ; « **\*toto\*** » tous les fichiers contenant « **toto** ».

Il faut également savoir que c'est le **shell** qui interprète ces caractères avant de transmettre la ligne de commande.

Par exemple, si vous tapez :

**rm t\*.tmp**

le shell transformera cette ligne de commande en :

**rm truc1.tmp truc2.tmp truc3.tmp**

## 2.6 Énoncé des tâches à réaliser sur le système :

Au moyen de **la ligne de commande** :

1. Créer l'arborescence de répertoires suivante dans votre répertoire personnel.

SecSy\_OS

- Linux
  - exercices
  - textes
- Rapports
- Python
  - exercices
  - textes
  - scripts

2. Afin de valider le point 1 de cet exercice, installez le package **tree** via la commande

**sudo apt-get install tree**

La commande **tree** vous permet d'afficher l'arborescence d'un dossier.

Essayez **tree ~/SecSy\_OS**

3. Dans le dossier SecSy\_OS→Rapports, éditez une synthèse des commandes utilisées dans le labo1 et 2 l'aide de l'éditeur nano
4. Déplacez-vous dans le répertoire Rapports, vérifiez que vous êtes dans le bon répertoire et affichez le contenu du répertoire Rapports :
5. Affichez le contenu du répertoire SecSy\_OS en utilisant le **chemin absolu**
6. Affichez le contenu du répertoire SecSy\_OS en utilisant le **chemin relatif**
7. Toujours sans vous déplacer du répertoire Rapports, affichez le contenu du répertoire Python en utilisant le **chemin absolu**
8. Affichez ensuite le contenu du répertoire Python en utilisant le **chemin relatif**
9. Comment faire pour revenir à votre répertoire \$HOME ?
10. Déplacez-vous dans le répertoire Python en utilisant le chemin relatif.
11. Puis retournez au répertoire Rapports en utilisant le chemin relatif.