

## TP4 - Linux: introduction à Bash

Objectifs :

- Le but de ce TP est de vous familiariser avec le shell Bash et quelques point clés d'une machine Linux.
- L'annexe qui vous est fournie vous résume quelques-unes des commandes utilisées dans ce TP, prenez le temps de prendre connaissance de son contenu avant de commencer le TP.
- Pendant le TP, se référer au deux cours : Shell et Systèmes de fichiers

### 1 Installation de la machine virtuelle

Pour ce TP, vous allez utiliser un système Debian 8 32 bits, via une machine virtuelle sous VirtualBox. Cette machine a été installée avec l'environnement de bureau Mate (on dit "mah-tay"), très populaire.

- Q1.1 - Lancez Oracle Virtualbox, puis Menu Fichier, Paramètres : vérifiez le dossier par défaut, qui doit être `c:\users\ (login) \.VirtualBox` avec (login) votre identifiant.
- Q1.2 - Copier dans ce dossier le dossier `P:\MV_VirtualBox\debian_8_32` (2,67 Go, ça peut être un peu long...)
- Q1.3 - Dans VirtualBox, menu "Machine ⇒ Ajouter", et aller chercher le fichier `debian_8_32.vbox` dans votre dossier :  
`c:\users\ (login) \.VirtualBox.`
- Q1.4 - Démarrez la machine, attendez le *boot* complet et loggez vous sur la machine, avec le login `user1`, password `user1`

### 2 Utilisation interactive

Lancer ensuite un terminal via le bureau : Menu "Applications→System Tools→Mate Terminal" : vous avez un shell à votre disposition.

#### 2.1 Commandes élémentaires

- Q2.1 - Quelle est la commande donnant le dossier courant :

- Q2.2 - Donner le dossier courant :

- Q2.3 - Avec `ls -l`, donner le nombre de fichiers et de dossiers existant dans le dossier

- Q2.4 - Donner la syntaxe de `ls` permettant de visualiser les fichiers cachés :

- Q2.5 - Votre configuration personnelle du terminal est stockée dans le fichier `.bashrc`. Afficher son contenu avec la commande `cat .bashrc`

Ce fichier est assez long, on peut en voir les dernières lignes avec la commande `tail`. En utilisant `man tail`, combien de lignes cette commande affiche-t-elle :

- Q2.6 - Donner la syntaxe de cette commande pour n'afficher que les 5 dernières lignes de `.bashrc` :

- Q2.7 - Pour compter les mots et lignes, Linux dispose de la commande `wc`. Par exemple, taper la commande `wc .bashrc` et donner les 3 valeurs affichées par la commande :

- Q2.8 - En vous aidant de `man wc`, à quoi correspondent les 3 valeurs :

- Q2.9 - A quoi sert l'option `-l` de la commande `wc` :

- Q2.10 - Pour compter le contenu d'un répertoire, on peut utiliser un "pipe" : une redirection de la sortie d'une commande vers l'entrée d'une 2<sup>e</sup>. Essayez par exemple `ls -l | wc -l`. Pourquoi la valeur affichée est différente de celle donnée à la question ???

Q2.11 - Déplacez vous dans le dossier **Documents** et y créez un fichier **toto** avec la commande `echo coucou > toto`. Vérifiez qu'il est bien présent avec **ls**.

Q2.12 - Donner la commande qu'on aurait du taper pour le créer au même endroit sans changer le dossier courant :

Q2.13 - Tapez `cd ~`. Que fait cette commande :

## 2.2 Gestion du Prompt (cf. annexe)

Q2.14 - Donner la commande pour afficher la variable d'environnement PS1 :

Q2.15 - Tapez les commandes suivantes `TOTO=$PS1`, puis `PS1="bonjour:"` Qu'obtenez-vous maintenant comme prompt :

Q2.16 - Ouvrez une seconde console. Avez vous le même prompt ?

Q2.17 - Revenez dans la 1<sup>re</sup> console et réinitialisez votre prompt avec

`PS1=$TOTO`

Q2.18 - En utilisant l'annexe, modifiez votre prompt afin d'obtenir l'affichage suivant : `xxx-yyy-10:23:42$`, avec `10:23:42` l'heure actuelle, `xxx` le nom de l'utilisateur courant, et `yyy` le chemin complet du répertoire courant. Donner la commande utilisée :

**Éditeur texte** : toutes les distributions fournissent en général au moins deux éditeurs texte : **vi** ("*Vee Aie*") et **nano**. Nous utiliserons ce dernier, plus simple d'approche (mais aussi moins puissant).

Q2.19 - Ouvrir le fichier de configuration **.bashrc** avec la commande

`nano .bashrc`

Rechercher la ligne `#force_color_prompt=yes` et la décommenter (enlever le caractère #).

Ajoutez en 1<sup>re</sup> ligne du fichier la commande `echo "Bienvenue !"`

Sauvegarder et quitter avec la séquence CTRL-O + Entrée + CTRL-X

Quitter la console avec `exit` et la relancer, et observez le changement.

**2.3 Alias** On peut créer des alias sur des commandes complexes, de façon à éviter les erreurs et à aller un peu plus vite.

Q2.20 - Tapez la commande **alias**. Combien sont définis par défaut :

Q2.21 - Tapez la commande `alias toto="echo toto !"` et essayez-là.

Apparaît-elle dans la sortie de **alias** ?

Q2.22 - Ouvrez un 2<sup>e</sup> terminal. L'alias est-il connu ?

Q2.23 - Dans le 1<sup>er</sup>, tapez la commande `unalias toto`, et visualisez la sortie de **alias**.

Q2.24 - Définissez l'alias **ll** comme un raccourci sur `ls -al`, en l'ajoutant dans votre **.bashrc**. Vérifier le fonctionnement, quel que soit le terminal utilisé.

## 2.4 Permissions sur les fichiers

Q2.25 - Visualisez le contenu de votre "home" et donner les permissions actuelles sur le fichier **toto**, ainsi que l'heure de création :

permissions :  heure :

Q2.26 - Donner la commande pour modifier les permissions pour ajouter aux autres membres du groupe le droit de modification.

Q2.27 - Vérifiez que vous pouvez toujours modifier le fichier avec la commande `touch toto`.

Que fait cette commande ? (utilisez `ls -l`)

### 3 Installation de nouveau paquets

L'explorateur de fichier graphique par défaut est un peu insuffisant pour un technicien. Vous allez installer un nouveau paquet : un autre explorateur de fichiers, appelé **nemo**.

Q3.1 - Lancer la commande **dpkg-query -f**, qui va lister tous les paquets installés.

Q3.2 - Proposez une commande pour afficher le nombre de paquets installés :

Q3.3 - Donner la commande utilisant la commande **grep** pour voir si le paquet **nemo** est dans la liste :

Q3.4 - La syntaxe pour installer un nouveau paquet est **apt-get install <nom-paquet>** Que se passe-t-il si vous essayez de l'utiliser avec l'utilisateur **user1** ? Pourquoi ?

Q3.5 - Installer **nemo** en passant "root" (mdp : root)

Q3.6 - Lancer **nemo** depuis le shell, et donner la version qui a été installée :

Q3.7 - Rappelez la commande de la question ??, et vérifiez que le paquet apparaît bien.

### 4 Gestion des utilisateurs

Q4.1 - Essayez de créer un compte **user2**. Pourquoi n'est-ce pas possible ?

Q4.2 - Passez en root et créez ce compte

Q4.3 - Visualisez l'ensemble des comptes existants sur la machine en affichant le fichier **/etc/passwd**.

Donner la commande permettant de compter le nombre d'utilisateurs :

Q4.4 - Actuellement les utilisateurs sont chacun dans leur propre groupe. Donner la commande permettant de créer un groupe **users** :

Q4.5 - A l'aide de l'annexe (partie 4), donner la commande pour affecter **user1** et **user2** au groupe **users** :

### 5 Création d'archive

La commande universelle pour gérer des archives dans l'environnement Linux est **tar**. Une archive est un fichier contenant un ensemble de fichiers avec leur chemin, ainsi que les métadonnées associées. On peut en plus avoir une compression non destructrice.

Cette commande est très utile, par exemple pour sauvegarder des données utilisateurs présente sur un serveur.

Syntaxe : **tar -[options] archive.tar [fichiers]**, avec

- **archive.tar** : nom du fichier archive
- [fichiers] est la liste des fichiers ou dossiers à archiver
- [options] étant une liste de commandes, parmi lesquelles
  - **f** (obligatoire) : indique que le 1<sup>er</sup> argument est le nom du fichier
  - **c** : création de l'archive
  - **v** ("*Verbose*") affiche la liste des fichiers au fur et à mesure
  - **x** : extraire les fichiers
  - **z** : ajout de la compression "zip"
  - **j** : ajout de la compression "bzip2" (taux de compression supérieur)

Exemples :

Création : **tar -cvf MesPhotos.tar /home/user1/Photos**

Extraction (dans le dossier courant) : **tar -xvf MesPhotos.tar**

#### Travail à effectuer

Q5.1 - Donner la commande pour archiver l'ensemble du dossier **/etc** dans un fichier **archive1.tar**, sans compression :

Q5.2 - Lancer cette commande en la préfixant avec la commande **time**, qui va mesurer la durée d'exécution.

Durée ("real") :  Taille archive :

Q5.3 - Donner la commande pour archiver l'ensemble du dossier **/etc** dans un fichier **archive1.tar.zip**, avec compression zip

Q5.4 - Lancer cette commande en la préfixant avec la commande **time**, qui va mesurer la durée d'exécution.

Durée ("real") :  Taille fichier :

Q5.5 - Donner la commande pour archiver l'ensemble du dossier **/etc** dans un fichier **archive1.tar.gz**, avec compression bzip2

Q5.6 - Lancer cette commande en la préfixant avec la commande **time**, qui va mesurer la durée d'exécution.

Durée ("real") :  Taille fichier :

Q5.7 - Mesurer le volume occupé par le dossier **/etc**, avec la commande **du** ("*Disk Usage*"), en précisant l'option **-h** ("*Human readable*"). Le total apparaît en bas.

Taille dossier sur disque :

Q5.8 - Comparer avec la taille du fichier **archive1.tar**. D'où vient la différence?

Q5.9 - Compléter le tableau ci-dessous :  
(taux compression = taille sans compression / taille avec compression)

	zip	bzip2
Taux compression		
Coeff. durée		

## 6 Recapitulatif

Récapitulez dans le tableau ci-dessus l'ensemble des commandes et outils que vous avez abordé dans ce TP (en dehors de ceux dédiés à la gestion des utilisateurs) :

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	