Introduction au système Linux

F. Lassabe

Printemps 2021

Contents

1 Rappels : manipulation des fichiers et dossiers

Cette première partie est un rappel ou une initiation à l'utilisation des commandes shell pour la manipulation du système de fichiers.

1.1 Chemins relatifs et absolus

La majorité des commandes de manipulation du système de fichiers prend en paramètre(s) des chemins d'accès aux dossiers et aux fichiers exposés par le système de fichiers. Le principe de Linux étant que tout est présenté sous forme de fichiers, ceux ci sont organisés sous la forme d'une arborescence, c'est-à-dire d'une structure partant d'une racine, et permettant d'accéder à l'ensemble de son contenu. Dans les systèmes d'exploitation basés sur Unix, dont Linux fait partie, la racine du système de fichiers est notée / (qui se prononce "slash"). Toute l'arborescence part de cette racine.

Un chemin est une manière de "dire" au système d'exploitation quel fichier ou dossier on veut manipuler. On distingue deux types de chemins :

- Les chemins absolus
- Les chemins relatifs

Les chemins absolus commencent par le caractère / et indiquent que le chemin doit être suivi depuis la racine du système de fichiers. Par exemple `/usr/include/c++/10/vector` est un chemin qui indique que le fichier **vector** est situé dans le répertoire **10**, lui même dans le répertoire **c++**, lui même dans le répertoire **include**, lui même dans **usr**, **usr** étant à la racine du système de fichiers.

Les chemins relatifs sont quant à eux exprimés à partir du dossier courant, c'est-à-dire le dossier dans lequel on est positionné au moment d'exécuter la commande. Ce dossier peut être connu grâce à la commande `pwd`, ou en affichant la variable d'environnement PWD (`echo $PWD`). Un chemin relatif commence par le nom d'un dossier, ou d'un fichier, mais ne commence en aucun cas par un caractère /. Par exemple, le chemin `Images/photo.png` représentera le chemin vers `/home/my_user/Images/photo.png` si le chemin courant est `/home/my_user`, le chemin `/var/www/html/Images/photo.png` si le chemin courant est `/var/www/html`, et ainsi de suite. Cela signifie qu'un chemin relatif a une cible qui varie suivant le dossier à partir duquel il est évalué.[1]

Il existe également, dans chaque dossier, deux dossiers cachés, l'un nommé "." qui représente le répertoire lui-même, un second nommé ".." qui représente le répertoire parent (celui dans lequel est situé le répertoire courant). Ceux-ci permettent notamment l'exécution d'un fichier exécutable présent dans le répertoire courant, et l'accès aux répertoires supérieurs.

1.2 Les commandes de base pour les répertoires

1.2.1 Déplacement dans l'arborescence

La première commande nécessaire est de pouvoir changer de répertoire courant. Celle-ci est la commande `cd` qui est un abrégé de *change directory*. Cette commande prend en paramètre un chemin. S'il est valide (i.e. le chemin existe), la commande change le répertoire courant. Si le chemin est invalide (il n'existe pas, ou correspond à un fichier), la commande échoue et affiche une erreur. Comme pour toutes les commandes pour manipuler le système de fichiers, le chemin peut être absolu ou relatif.

À partir de cette introduction, répondre aux questions suivantes :

- Depuis le répertoire courant `/home/users/bob/Images`, quelle commande vous permettra d'aller dans le répertoire `/home/users/bob` ?

- Depuis le répertoire courant `/home/users/bob/Images`, quelle commande vous permettra d'aller dans le répertoire `/var/log` ?

[1] On parle d'évaluer un chemin quand le système reconstruit le chemin absolu à partir du chemin relatif

Parmi les chemins suivants, lesquels sont absolus, lesquels sont relatifs ?

Chemin	Absolu	Relatif
/var/log		
Vidéos/FIMU		
/etc/fstab		
Documents/Super.Cours.de.LP25		

1.2.2 Création de répertoires

La seconde commande nécessaire est celle qui permet de créer un répertoire, `mkdir` (pour *make directory*). Celle ci prend en paramètre un chemin vers le dossier à créer. Le dernier élément doit être un chemin **qui n'existe pas encore**. Par exemple :

Commande	Effet
mkdir Rep	Crée le répertoire Rep dans le répertoire courant
mkdir /var/log/Rep	Crée le répertoire Rep dans le dossier /var/log
mkdir -p Rep1/Rep3/Rep6	Crée le répertoire Rep6 dans Rep1/Rep3 (Rep1 étant dans le répertoire courant). L'option -p permet de créer Rep1 et Rep3 si ceux ci n'existent pas encore.

1.2.3 Suppression de répertoires

Inversement, `rmdir` (*remove directory*) permet de supprimer un répertoire **à la condition que ce dernier soit vide**[2]. Cette commande échoue si le chemin en paramètre est invalide (il référence un chemin inexistant, ou un chemin vers un fichier). Il est également possible de supprimer un répertoire et tout son contenu avec la commande `rm -rf CHEMIN`. Attention, cette dernière est dangereuse, car une fausse manipulation peut vous amener à détruire un répertoire avec des données importantes puisqu'elle supprime le répertoire et tout ce qu'il contient (fichiers et sous-répertoires et leurs contenus).

1.2.4 Listage du contenu des répertoires

On utilise la commande `ls` pour lister le contenu d'un répertoire (qui peut être dans le répertoire courant, ou dans un chemin plus éloigné). Cette commande affiche par défaut uniquement les noms des répertoires et fichiers standards. Elle dispose d'un grand panel d'options dont les plus couramment utilisées sont les suivantes :

-a affiche également les fichiers cachés

-l affiche des informations détaillées (dont les droits d'accès, le propriétaire, la taille, la date de modification)

-h affiche la taille dans une unité facile à lire (octets, kilo-octets, méga-octets, etc.)

Sans argument, la commande ls va lister le contenu du répertoire courant. On peut également lui passer en argument le chemin du répertoire que l'on veut visualiser.

À partir des données ci-dessus, proposez les commandes permettant de répondre aux besoins suivants :

- Lister le contenu du répertoire courant, fichiers cachés compris

- Lister le contenu du répertoire /var/www avec les informations détaillées

- Lister le contenu du répertoire courant, avec les informations détaillées, des tailles plus faciles à lire, et les fichiers et dossiers cachés.

1.3 Les commandes de base pour les fichiers

Tout comme les répertoires, les fichiers peuvent être manipulés mais n'offrent pas les mêmes commandes.

[2] À l'exception des répertoires . et ..

1.3.1 Création d'un fichier

Il n'existe pas une façon unique de créer un fichier (en effet, la création des fichiers est généralement le fait d'une application qui gère son propre format en fonction de ses fonctionnalités). Cependant, pour les fichiers de texte brut en ASCII ou en unicode, il est possible de créer un fichier vide sans entrer dans un éditeur grâce à la commande `touch`. Cette commande a pour rôle de mettre à jour la date de modification d'un fichier à la date et l'heure où la commande touch a été exécutée. Un effet particulier de cette commande est que, si le fichier en paramètre n'existe pas, elle va le créer (en générant un fichier vide).

1.3.2 Copie

La commande `cp` permet de copier un fichier. Elle prend comme premier paramètre le chemin vers la source, et comme second paramètre le chemin vers la destination. Le chemin de destination peut être un nom différent, ce qui permet à cp de copier en renommant, comme vous le verrez dans les exemples suivants :

Commande	Effet
cp fichier.txt fichier_copie.txt	Crée une copie de fichier.txt (qui est dans le rép. courant), nommée fichier_copie.txt (également dans le rép. courant)
cp fichier.txt Documents/	Copie fichier.txt dans le répertoire Documents (notez le / à la fin du nom pour identifier qu'il s'agit d'un répertoire), en conservant son nom
cp fichier.txt Documents/fichier_backup.txt	Copie fichier.txt sous le nom fichier_backup.txt dans le répertoire Documents
cp -r Documents/ Documents_backup	Copie le répertoire Documents **et son contenu** sous le nom Documents_backup
cp -r Documents/ /media/usbdrive/Documents_backup	Copie le répertoire Documents **et son contenu** sous le nom Documents_backup dans le répertoire /media/usbdrive

Comme vous pouvez le voir avec les exemples, cp s'applique également aux répertoires.

1.3.3 Renommage et déplacement

La commande `mv` permet de déplacer et/ou de renommer fichiers et répertoires. Elle prend deux arguments : le chemin vers la source, et le chemin vers la cible. Si la source est un fichier, la cible peut être un dossier ou un fichier. Si la source est un dossier, la cible ne peut qu'être un dossier.

Si la source est un fichier, et que la cible est un fichier, la source est renommée. Si la source est un fichier et que la cible est un dossier, la source est simplement déplacée. Enfin, si la source est un dossier, le résultat dépend de la cible : si c'est un dossier qui existe, le dossier y est déplacé, sinon il est renommé (et éventuellement déplacé).

Commande	Effet
Avec des fichiers	
mv fichier1 Documents	Déplace fichier1 dans Documents (si c'est un rép.), le renomme en "Documents" sinon
mv fichier1 /var/www/index.html	Déplace fichier1 dans /var/www en le renommant en index.html
Avec des répertoires	
mv Rep1 Rep2	Renomme Rep1 en Rep2 si Rep2 n'existe pas, déplace Rep1 dans Rep2 sinon
mv Rep1 /var/www/mon_site	Renomme Rep1 en mon_site en le déplaçant dans /var/www si /var/www/mon_site n'existe pas, déplace Rep1 dans /var/www/mon_site sinon

1.3.4 Suppression

La commande `rm` est utilisée pour supprimer des fichiers. Elle prend en paramètre une ou plusieurs cibles à supprimer. Elle évalue toutes les cibles et supprime celles qui sont valides. Elle affiche une erreur pour les chemins invalides.

2 Informations système

2.1 Principales commandes d'informations système

À l'aide des commandes suivantes :

- `uname`
- `echo \$OSTYPE`
- `uname -a`
- `cat /proc/version`
- `cat /proc/cpuinfo`
- `df [-hT]`
- `mount`
- `ip addr`
- `ip route`

Répondez aux questions suivantes :

1. Quel est le système d'exploitation installé sur votre machine ?

2. De combien de processeurs dispose votre machine ?

3. Quel est le type et la fréquence du processeur ?

4. Visualiser les fichiers /etc/mtab et /etc/fstab, laquelle des commandes ci-dessus tire ses résultats du contenu de ces fichiers ?

5. Que signifie ext4, ntfs, tmpFS, Sysfs ?

6. Sur quel périphérique se trouve le système de fichiers Linux et quel est son point de montage ?

7. De quel type d'interface réseau dispose votre machine, quelle est son adresse IP, et quelle est sa route par défaut ?

3 Principaux répertoires du système Linux

Une brève description du contenu des principaux répertoires du système linux est faite ci-dessous. À l'aide de la commande `ls`, visualiser le contenu des répertoires décrits ci dessous, et répondre aux questions.

/bin contient les commandes de base

/sbin contient les commandes nécessaires à l'administration système

/boot contient les informations nécessaires au démarrage de la machine dont les fichiers du noyau Linux. Vérifiez que la version du noyau trouvée dans le premier exercice a ses fichiers dans /boot, en écrivant dans la zone ci dessous les noms des fichiers relatifs au noyau actuellement en fonctionnement :

/dev Linux accède aux éléments matériels (périphériques) de la machine par l'intermédiaire de fichiers spéciaux contenus dans le répertoire /dev

/etc contient les fichiers de configuration du système.

/home contient les répertoires personnels des utilisateurs.

/lib contient les principales bibliothèques partagées utilisées pour le développement.

/mnt ou /media contient les répertoires utilisés pour monter les partitions et les périphériques amovibles (USB, CD, disquette ...).

/proc un répertoire en mémoire, dont les fichiers contiennent des infos sur l'état du système et des processus en cours d'exécution.

/root le répertoire de l'administrateur système.

/tmp contient les fichiers temporaires.

/usr Principal répertoire du système :

- **/usr/bin** les fichiers exécutables installés avec le système.
- **/usr/include** les fichiers d'en-tête pour la programmation.
- **/usr/share/man** le manuel en ligne.

/var contient les parties variables du système (des données fréquemment réécrites) comme les journaux de l'activité du système (logs), les files d'attentes pour les impressions, les boites aux lettres, etc.

Vérifiez la localisation des commandes utilisées dans l'exercice 1 :

	/bin	/sbin	/usr/bin
uname			
echo			
cat			
df			
mount			
ip			

4 Gestion des utilisateurs

4.1 Identification des utilisateurs du système

Lire les pages de manuel des commandes suivantes : **whoami**, **who**, **id** et **passwd**.

- Que fait la commande `whoami` ?
- Que vous renvoie la commande `whoami` ?
- Que fait la commande `who` ?
- Que vous renvoie la commande `who` ?
- Que fait la commande `id` ?
- Que vous renvoie la commande `id` ?
- Comment utiliser la commande `passwd` pour changer votre mot de passe ?

4.2 Données des comptes utilisateurs

Le compte d'un utilisateur est représenté par son *login* et un mot de passe associé. Les informations sur les comptes utilisateurs du système Linux sont regroupées dans le fichier **/etc/passwd**. Chaque ligne de ce fichier correspond à un compte. Une ligne est composée de 7 champs séparés par des ":". Les champs sont les suivants :

1. Le nom du compte de l'utilisateur 2.

2. Un x à la place du mot de passe.
3. Un entier qui identifie l'utilisateur pour le système d'exploitation (UID=User ID, identifiant utilisateur)
4. Un entier qui identifie le groupe de l'utilisateur (GID=Group ID, identifiant de groupe)
5. Le commentaire dans lequel on peut retrouver des informations sur l'utilisateur comme son nom réel
6. Le répertoire personnel de l'utilisateur dans lequel il se trouve après s'être connecté au système
7. La commande exécutée après connexion au système (par exemple l'interpréteur de commandes **bash**)

Les informations concernant les groupes sont conservés dans le fichier **/etc/group**. Ce fichier contient une entrée par groupe. Chaque entrée est composée de 4 champs :

1. Le nom du groupe
2. Mot de passe du groupe qui est rarement utilisé
3. L'entier qui identifie le groupe pour le système d'exploitation (GID=Group ID, identifiant groupe)
4. Liste des membres qui est la liste des login des utilisateurs membres du groupe. Les utilisateurs dont c'est le groupe principal n'ont pas besoin d'apparaître dans cette liste.

Tout utilisateur du système peut lire les fichiers **passwd** et **group** pour, par exemple, connaître le nom d'un autre utilisateur.

- Ouvrir et visualiser le contenu des deux fichiers. Est-ce qu'on peut lire le mot de passe des utilisateurs ?
  Oui Non
- En réalité, le mot de passe de l'utilisateur est contenu dans le fichier **/etc/shadow**. Essayez de lire ce fichier, que se passe-t-il ?