

L'administration d'un serveur Apache sous Linux

Module 2 - Le système de fichiers



Objectifs

- Apprendre à se déplacer dans l'arborescence
- Savoir manipuler des fichiers

Le FHS

- Le **FHS** (*Filesystem Hierarchy Standard*) a été créé en 1993 pour harmoniser l'arborescence d'un système de fichiers Linux/Unix
- La version 3.0 actuelle date de juin 2015
- Ce standard est maintenu par la **Linux Foundation**
- Il est important de connaître les dossiers pour savoir retrouver les fichiers

La racine

- Le système de fichiers GNU/Linux se présente sous forme d'arborescence
- La **racine** notée « / » est le début de cette arborescence
- **Ne pas l'oublier**

Les répertoires *bin

/bin	Commandes de base nécessaires au démarrage et à l'utilisation minimale du système pour tous les utilisateurs
/sbin	Exécutables système pour les administrateurs
/usr/bin	Binaires non indispensables à un système minimal. Répertoire d'installation des applications installées
/usr/sbin	Binaires d'administration

Les répertoires système

<code>/boot</code>	Répertoire pour le chargeur d'amorçage et fichiers nécessaires au lancement du noyau.
<code>/etc</code>	Fichiers de configuration du système et des services
<code>/lib</code>	Bibliothèques principales utilisées par les commandes de <code>/bin</code> et <code>/sbin</code>
<code>/media</code> et <code>/mnt</code>	Points de montage des périphériques de stockage amovibles (<code>/media</code>) et temporaires (<code>/mnt</code>)

Les répertoires système

/opt	Répertoire pour les applications tierces
/usr	Contient certains répertoires présents à la racine mais qui ne sont pas nécessaires au fonctionnement minimal du système.
/var	Fichiers de taille variable : Temporaires : /var/cache , /var/run Journaux : /var/log Web : /var/www

Le système de fichiers

Les répertoires utilisateurs

<code>/home</code>	Répertoire d'accueil des utilisateurs ou répertoire personnel des utilisateurs
<code>/root</code>	Répertoire personnel de l'utilisateur root

Le système de fichiers

Les répertoires « virtuels »

<code>/dev</code>	Fichiers de périphériques physiques (disque dur, etc.) ou virtuels (<code>/dev/null</code> , <code>/dev/zero...</code>)
<code>/proc</code>	Système de fichiers virtuel reflétant l'état du noyau et les différents processus

Les types de fichiers

- **Sous Unix, tout est fichier**
- L'utilisateur « voit » les fichiers **organisés de façon hiérarchique**
- Le système « voit » les fichiers comme **des données sur un support** accessible via un **inode** pointant vers les données
- L'accès aux périphériques se fait aussi via des fichiers spécifiques

Le système de fichiers

Les types de fichiers

- Il existe quatre types de fichiers :
 - Fichiers standards : textes, scripts, images...
 - Répertoires : contient les références permettant d'accéder aux fichiers
 - Liens symboliques : fichiers spéciaux permettant de pointer vers un nom de fichier
 - Les fichiers spéciaux

Le système de fichiers

Les types de fichiers

- Il existe deux modes pour les fichiers spéciaux :
 - Fichiers spéciaux en mode bloc : pour les périphériques de stockage
 - Fichiers spéciaux en mode caractère ou raw : pour les périphériques d'entrée-sortie (clavier, souris...)

Désignation des fichiers

- La notion d'**extension** de fichier sous Linux n'existe pas. Elle est utile pour l'utilisateur et les applications pour reconnaître le type de fichier.
- Pas de caractères interdits sauf le « / », mais attention aux caractères spéciaux difficilement manipulables.
- Le nom d'un fichier est limité à 255 caractères.

Désignation des fichiers

- Linux est **sensible à la casse** : fichier.txt et Fichier.txt sont deux fichiers différents
- Les fichiers commençant par un « . » sont des fichiers cachés lors de l'exécution de certaines commandes et sont cachés dans certaines applications
- Les fichiers cachés servent souvent pour stocker des configurations

Le système de fichiers

Les chemins

- On parle de **chemin absolu** quand on décrit l'accès à un fichier depuis la racine /
- On parle de **chemin relatif** quand on décrit l'accès à un fichier depuis le **répertoire courant**
- Le caractère « . » désigne **le répertoire courant**
- Le caractère « .. » désigne **le répertoire parent**

Le système de fichiers

La commande pwd

- Syntaxe : `pwd`
- Commande permettant d'afficher le répertoire dans lequel l'utilisateur est positionné

La commande cd

- Syntaxe : `cd <chemin>`
- Commande permettant de naviguer dans le système de fichiers
- `<chemin>` désigne le répertoire dans lequel on veut se positionner
- Exemples :
 - `$ cd /var/log`
 - `$ pwd`
 - `$ cd ../spool`

Le système de fichiers

La commande cd

- Quelques raccourcis :
 - Revenir au répertoire précédent : `$ cd -`
 - Revenir au répertoire de l'utilisateur : `$ cd` ou `$ cd ~`

La commande ls

- Syntaxe : `ls [options] <répertoires ou fichiers>`
- Commande permettant de lister les fichiers présents ou de lister les infos sur les fichiers
- Options intéressantes :
 - `-a` : affiche tous les fichiers
 - `-A` : idem `-a` sauf `.` et `..`
 - `-l` : affichage au format long
 - `-d` : affiche les infos d'un répertoire
 - `-h` : affichage des tailles en mode « humain »

La commande ls

- Exemples :
 - `$ ls /etc`
 - `$ ls -lah /etc`
 - `$ ls -lahd /etc`
- Remarquez la différence entre les deux dernières commandes

Le système de fichiers

Les informations affichées

- **c1** : désigne le type de fichier
 - - : fichier normal
 - d : répertoire
 - l : lien symbolique
- **c2-c10** : droits sur le fichier
- **c11** : nombre de liens physiques pour un fichier ou nombre de sous-répertoires pour un répertoire
- **root root** : propriétaire et groupe propriétaire
- **19605** : taille en octets
- **oct. 25 2014** : heure ou date de dernière modification

Le système de fichiers

Se déplacer sur le système de fichiers

Démonstration



La commande touch

- Syntaxe : `touch <fichier>`
- Commande permettant de créer des fichiers vides
- Exemple : `$ touch fichier1.txt`

La commande mkdir

- Syntaxe : `mkdir [options] <répertoire>`
- Commande permettant de créer des répertoires
- Options intéressantes :
 - `-m` : crée un répertoire en affectant les droits
 - `-p` : crée le répertoire parent si inexistant
- Exemples :
 - `$ mkdir -p Eni/Formation/Apache`
 - `$ mkdir -m 770 Eni/Formation/Apache/Tp`

Le système de fichiers

Créer des répertoires et des fichiers

Démonstration



La commande cp

- Syntaxe : `cp [options] <source> <cible>`
- Commande permettant de copier un fichier ou un répertoire
- Options intéressantes :
 - `-r` : copie récursive (copier un répertoire et son contenu)
 - `<source>` : fichier à copier
 - `<cible>` : le nom du nouveau fichier ou un autre répertoire

La commande cp

- Exemples :
 - `$ cp /etc/services ./`
 - `$ cp -r /etc ./backup_etc/`
- Remarquez que sur la seconde commande le répertoire `./backup_etc` est créé en même temps

La commande mv

- Syntaxe : `mv [options] <source(s)> <cible>`
- Commande permettant de renommer et/ou de déplacer des fichiers
- Exemples :
 - `$ mkdir tmpeni`
 - `$ touch fic{1,2,3}`
 - `$ mv fic1 fic11`
 - `$ mv fic* tmpeni/`

La commande rm

- Syntaxe : `rm [options] <fichier ou répertoire>`
- Commande permettant de supprimer un fichier ou un répertoire
- Options intéressantes :
 - `-r` : suppression récursive
 - `-i` : demande de confirmation
- Exemples :
 - `$ rm ./backup_etc/services`
 - `$ rm -rf ./backup_etc/`

Le système de fichiers

Copier ou déplacer des fichiers

Démonstration



Conclusion

- Vous savez vous y retrouver dans l'arborescence du système de fichiers
- Vous savez dans quel répertoire placer les différents fichiers (configuration, données, etc.)