

Chapitre II : Introduction Au Système d'exploitation Linux

I. Généralités

1) *Qu'est ce qu'un système UNIX ?*

UNIX est un système d'exploitation écrit en 1969 par Ken Thompson dans les laboratoires de Bell. La plupart d'UNIX est écrit dans le langage de programmation C, il est donc capable de fonctionner sur une large gamme d'architectures informatiques.

I. Généralités

2) Le shell

Le Shell (en français, la coquille ou interpréteur de commande) est l'interface utilisateur sous Unix. Lorsqu'un utilisateur tape des commandes Unix, ces commandes sont reçues par le Shell qui les interprète avant de lancer l'exécution de cette commande. Le Shell est une couche logicielle bien séparée du noyau. Il joue un double rôle celui d'interpréteur de commandes et celui de langage de programmation.

L'utilisateur, à l'aide des commandes qu'il a à sa disposition, peut écrire ses propres fonctions et programmes en langage Shell, ce sont alors des shellscripts. Une fois ceux-ci réalisés, ils peuvent être utilisés par l'utilisateur comme n'importe quelle commande du Shell lui même.

I. Généralités

3) *Caractéristiques du système Linux*

Les caractéristiques du système Linux sont :

- ▶ **Système multi-utilisateurs:** Linux permet de gérer plusieurs utilisateurs à la fois;
- ▶ **Système multitâches:** Il permet l'exécution de plusieurs programmes simultanément, grâce à l'environnement X-Windows (KDE ou gnome = interface graphique rendant l'interaction avec le système plus conviviale)
- ▶ **La sécurité** Linux offre de bonnes garanties de sécurité. car, il est conçu dès le départ pour fonctionner en réseau. Les bogues découverts sont corrigés très rapidement.
- ▶ **Multi-plate-formes** (c'est-à-dire portable)

I. Généralités

4) *Notions de noyau et de distribution*

Linux est architecturé autour d'un **noyau** (en anglais **kernel**) chargé de prendre en charge le matériel. On appelle **distribution** l'assemblage d'un ensemble de logiciels autour d'un noyau Linux afin de fournir un système clé en main. Le noyau d'une distribution peut-être mis à jour afin de permettre la prise en compte de matériels récents, toutefois cette manipulation consistant à recompiler le noyau est délicate car elle nécessite un certain niveau de connaissance du système et du matériel.

I. Généralités

4) *Notions de noyau et de distribution*

- ▶ Chaque distribution possède ses avantages et ses inconvénients. Les distributions les plus connues sont :
- ▶ *La distribution RedHat ;*
- ▶ *La distribution Ubuntu*
- ▶ *La distribution Debian ;*
- ▶ *La distribution SuSe ;*
- ▶ *La distribution Knoppix ;*
- ▶ *La distribution Slackware ;*
- ▶ *La distribution Mandriva;*

I. Généralités

5) *Arborescence des dossiers linux*

Sous Linux (et sous UNIX en général), vous êtes dans une arborescence. C'est à dire qu'il y a une organisation hiérarchique de répertoires, qui peuvent contenir des fichiers et/ou des répertoires.

Cette notion est simple, puisque le monde Microsoft l'a repris. Par contre, il n'y a pas de C: ou de D:. Dans le monde Windows, il y a une arborescence pour chaque partition. Sous Linux, Il n'y a qu'une et une seule arborescence.

I. Généralités

5) *Arborescence des dossiers linux*

Liste des principaux répertoires de l'arborescence Linux :

- ▶ **bin** : contient des programmes (exécutables) qui sont susceptibles d'être utilisés par tous les utilisateurs de la machine.
- ▶ **boot** : dossier permettant le démarrage de Linux.
- ▶ **dev** : répertoires contenant les périphériques. il contient des sous-dossiers dont chacun représente un périphérique. On y retrouve ainsi par exemple le fichier qui représente le lecteur CD.
- ▶ **etc** : répertoires de configuration.

I. Généralités

5) *Arborescence des dossiers linux*

- ▶ **home** : répertoires personnels des utilisateurs. c'est dans ce dossier que vous placerez vos fichiers personnels, à la manière du dossier "Mes documents" de Windows. Chaque utilisateur de l'ordinateur possède son dossier personnel.
- ▶ **lib** : dossier contenant les bibliothèques partagées (généralement des fichiers .so) utilisées par les programmes. C'est en fait là qu'on trouve l'équivalent des .dll de Windows.
- ▶ **media** : lorsqu'un périphérique amovible (comme une carte mémoire SD ou une clé USB) est inséré dans votre ordinateur, Linux vous permet d'y accéder à partir d'un sous-dossier de "media". On parle de montage.

I. Généralités

5) *Arborescence des dossiers linux*

- ▶ **root** : c'est le dossier personnel de l'utilisateur "root". Normalement, les dossiers personnels sont placés dans "home". mais celui de root fait exception.
- ▶ **sbin** : contient des programmes système importants.
- ▶ **tmp** : dossier temporaire utilisé par les programmes pour stocker des fichiers.
- ▶ **usr** : c'est un des plus gros dossiers, dans lequel vont s'installer la plupart des programmes demandés par l'utilisateur.

II. ***Les commandes Linux***

1) Introduction

Pour travailler avec UNIX, il est important de savoir qu'il est possible de communiquer directement avec le noyau du système. Cette communication passe par le SHELL qui sert d'intermédiaire entre la saisie des commandes et le noyau du système.

La syntaxe générale d'une commande linux est :

Commande [options] paramètres

- ▶ En général, les options sont précédées du symbole - et peuvent être groupées;
- ▶ Les paramètres précisent en général les fichiers concernés

II. *Les commandes Linux*

2) **Démarrage et arrêt de session**

2.1) Commande *reboot*

La commande reboot permet de redémarrer de la machine

2.2) Commande *shutdown*

Permet d'arrêter et redémarrer la machine.

Par exemple :

II. *Les commandes Linux*

2) Démarrage et arrêt de session

- ▶ **shutdown** : arrêter proprement le système.
- ▶ **shutdown +15**: arrêter proprement le système après 15 minutes.
- ▶ **shutdown 15 :00**: arrêter proprement le système (rebooter) à 15 :00 minutes.
- ▶ **shutdown -r** : redémarrer le système (rebooter).
- ▶ **shutdown -r +15**: redémarrer le système (rebooter) après 15 minutes.
- ▶ **shutdown -r 15 :00**: redémarrer le système (rebooter) à 15 :00 minutes.

II. *Les commandes Linux*

3) **Commandes de répertoires**

3.1) Commande *pwd*

La commande `pwd` (**Print Working Directory**), cette commande vous **indique le nom du répertoire actif**, c'est à dire celui dans lequel vous êtes en train de travailler.

Exemple d'utilisation :

```
pwd
```

II. *Les commandes Linux*

3) **Commandes de répertoires**

3.2) Commande ls

La commande ls permet d'afficher le contenu du répertoire courant (ou du répertoire demandé).
Equivalent de DIR sous Dos.

Exemples

- ▶ **ls** : Permet d'afficher le contenu du répertoire courant
- ▶ **ls /** : Permet d'afficher le contenu du répertoire racine
- ▶ **ls Bureau** : permet d'afficher le contenu du répertoire Bureau

II. *Les commandes Linux*

3) **Commandes de répertoires**

On peut utiliser d'autres options avec la commande ls.
Par exemples :

- ▶ **ls -a** : permet d'afficher **les fichiers cachés**.
- ▶ **ls -F** : Affiche les fichiers par type. Ainsi un fichier suivi d'un slash (/) est un répertoire, un fichier suivi d'une **étoile** est un fichier exécutable et un fichier suivi d'un "@" est un lien.
- ▶ **ls -m** : Affiche les fichiers en les séparant par des virgules au lieu de les présenter en colonnes.

II. *Les commandes Linux*

3) **Commandes de répertoires**

- ▶ **ls -t** : Affiche les fichiers par date, c'est-à-dire en les classant du récent au plus ancien.
- ▶ **ls -lu** : Affiche les fichiers par date de dernier accès et indique cette date.
- ▶ **ls -r** : Affiche les fichiers en ordre alphabétique inverse
- ▶ **ls -S** : Affiche les fichiers triés par ordre décroissante de taille.
- ▶ **ls -l** : Présenter un fichier par ligne.

II. *Les commandes Linux*

3) **Commandes de répertoires**

3.3) *Commande mkdir*

La commande **mkdir** permet la création d'un ou plusieurs répertoires dont le chemin est passé en argument.

Exemple :

- ❑ **mkdir rep1** : Cette commande permet de créer le répertoire **rep1** dans le dossier courant.
- ❑ **mkdir rep1/rep11** : Cette commande permet de créer le répertoire **rep11** dans le répertoire **rep1** du dossier courant.

Pour créer une hiérarchie on doit utiliser l'option **-p** qui permet de créer les dossiers intermédiaires:

- ❑ **mkdir -p rep2/rep12/rep112** : Cette commande permet de créer les répertoires **rep2**, **rep12**, et **rep112** (en respectant la même arborescence)