

NOTES DE COURS
ABDOULAYE DABO

Département de Genie Informatique (DGI)
Ecole supérieure polytechnique (ESP)
Université Cheikh Anta Diop (UCAD)

Fondamentaux des systèmes d'exploitation de
type UNIX/LINUX

Table des matières

1	Introduction	2
2	Historique	2
3	Système de fichier	2
4	Le terminal et le shell	3
5	Syntaxe de base d'une commande linux	3
6	Scripting shell	4
6.1	Structure d'un script shell	4
6.2	Comment rendre un script exécutable?	4

1 Introduction

Un ordinateur est formé : d'une partie **matérielle** (UC, mémoire, disques, terminaux CPU, RAM, disque, écran, clavier, etc) et d'une partie **logicielle**. Ce dernier est constitué de programmes utilisateurs et de programmes systèmes. Le **systèmes d'exploitation (Operating System en Anglais)** est l'un des principaux programme systèmes.

Le SE est ensemble de programme qui dirige l'utilisation des ressources (gestion de processus, de la mémoire, système de fichiers, E/S) de l'ordinateurs par des logiciels applicatifs.

Il sert d'**intermédiaire** entre les logiciels et le matériel à travers des **appels systèmes**.

Exemples de SE : Windows, Unix, MacOS,...

2 Historique

Le noyau **Linux** a été conçu par **Linus Torvalds en 1992** . Une **distribution** Linux est un ensemble qui se compose d'un noyau Linux et des logiciels issus du projet GNU.

Exemples de distribution Linux : Debian, Ubuntu, Fedora, Redhat,...

3 Système de fichier

L'organisation traditionnelle du répertoire racine est décrite dans le tableau suivante :

Répertoire	Contient
/bin	les fichiers exécutables nécessaires à l'initialisation
/boot	le noyau et les fichiers de démarrage
/dev	les fichiers spéciaux et périphériques
/etc	les fichiers de configuration du système
/home	la base des répertoires utilisateurs
/lib	les librairies système et les modules
/mnt	les points d'ancrage des systèmes extérieurs
/proc	un système de fichiers virtuels permettant l'accès aux variables du noyau
/root	le répertoire de base du super utilisateur
/sbin	les commandes d'administration
/tmp	les fichiers temporaires
/usr	les programmes, les librairies et les fichiers accessibles pour l'utilisateur
/var	les données variables liées à la machine

Quelques répertoires importants

- / est la racine (à peu près comme **C :**\ sous Windows).
- . est le répertoire actuel
- .. est le répertoire parent
- /home/abdoulaye désigne le répertoire personnel de l'utilisateur abdoulaye
- ~ désigne répertoire personnel de l'utilisateur courant

Linux étant un système multi-utilisateurs, chaque utilisateur dispose de son propre répertoire de connexion.

4 Le terminal et le shell

Le **Shell** (interface d'appels systèmes) est l'interpréteur en ligne de commande, il permet à un utilisateur de s'identifier sur un terminal. Il fournit un certain nombre de commandes pour interagir avec le système d'exploitation. C'est aussi un langage de programmation. En clair donc, lorsque vous utilisez Linux, vous ouvrez un terminal pour accéder à un Shell afin de passer des commandes.

Exemples de shell :

Sur Linux : Bourne shell (sh), KornShell (ksh), C shell (csh), tcsh ,bash, Z shell (zsh)

Sur Windows : PowerShell, CMD

5 Syntaxe de base d'une commande linux

nom_commande [Options] Arguments

Les options ne sont pas obligatoires. Cependant elles permettent d'utiliser toute la puissance des commandes.

Elles peuvent être combinées comme suit :

nom_commande -option1option2 arguments

Exemple

ls -l /home : Cette commande permet de lister le contenu du répertoire /home

ls -la /home : l'option **-a** permet d'afficher les fichiers cachés du répertoire /home

Les commandes suivantes seront utiles :

- **cd** : permet d'aller dans un répertoire
- **ls** : affiche le contenu d'un répertoire
- **pwd** : donne le chemin vers le répertoire actuel

- **mkdir** : permet la création d'un ou plusieurs répertoires
- **rmdir** : permet la suppression d'un ou plusieurs répertoires
- **man commande** : permet d'avoir accès à l'aide sur cette commande :
- **apropos commande** : affiche une ligne d'information sur l'objectif de la commande .
- **cp origine arrivée** : copie les fichiers.

Origine peut être un ou plusieurs fichiers arrivé peut être un fichier (si une seule origine), ou un répertoire (si une ou plusieurs origines).

- **mv** : comme cp, si ce n'est qu'il efface l'origine
- **rm** : suppression du ou des fichiers donnés en argument
- **touch** : permet la création d'un ou plusieurs fichiers

6 Scripting shell

Le shell est une interface qui permet à l'utilisateur d'interagir avec le système d'exploitation. Pour y accéder l'utilisateur ouvre un terminal et tape des commandes textuelles exécutées ensuite par le shell. Cependant pour réaliser plusieurs tâches en ligne de commande, on se voit être limité. Le scripting shell a pour objectif d'effectuer plusieurs tâches avec un seul script et d'automatiser des opérations répétitives.

6.1 Structure d'un script shell

```
#!/bin/bash
#Ecrire ici les instructions
echo "Hello World!"
```

- La première ligne du script indique le shell utilisé pour interpréter le contenu du fichier.
suivi du shell (ex : `#!/bin/bash`) Indique que le shell bash interprètera le fichier.
- Le caractère `"#"` délimite les commentaires dans un script shell. Tout ce qui suit le `"#"` sur une ligne est considéré comme un commentaire.
- **echo** permet d'afficher un message

6.2 Comment rendre un script exécutable ?

La variable d'environnement **\$PATH** contient les chemins vers les répertoires contenant les fichiers exécutables. Pour rendre notre script exécutable, il est donc nécessaire d'ajouter le chemin vers le répertoire le contenant au **\$PATH**.

```
$ echo 'export PATH=$HOME/chemin/vers/le/fichier :$PATH'
```

Mais cette modification n'existe que dans le shell courant. Pour la rendre définitive il faut l'écrire dans un fichier de configuration lu au démarrage (`~/.bashrc`, `~/.profile`).

```
$ echo 'export PATH=$HOME/chemin/vers/le/fichier :$PATH' » ~/.bashrc
```

Pour executer :

```
$ nom_fichier.sh
```