

## 1 Introduction

UNIX est un système d'exploitation multi-tache et multi-utilisateur.

Permet à un ordinateur mono ou multi-processeurs de faire exécuter simultanément plusieurs programmes par un ou plusieurs utilisateurs.

### 1.1 Le système de gestion de fichiers

Le système de fichiers racine (root file system) est associé au répertoire le plus haut /, il contient les sous répertoires suivants :

- **/bin** commandes binaires utilisateur essentielles (pour tous les utilisateurs)
- **/boot** fichiers statiques du chargeur de lancement
- **/dev** fichiers de périphériques
- **/etc** configuration système spécifique à la machine
- **/home** répertoires personnels des utilisateurs (optionnel)
- **/lib** bibliothèques partagées essentielles et modules du noyau
- **/mnt** point de montage pour les systèmes de fichiers montés temporairement
- **/proc** système de fichiers virtuel d'information du noyau et des processus
- **/root** répertoire personnel de root (optionnel)
- **/sbin** binaires système (binaires auparavant mis dans /etc)
- **/sys** état des périphériques (model device) et sous-systèmes (subsystems)
- **/tmp** fichiers temporaires

### 1.1 Qu'est-ce que le shell ?

C'est l'interpréteur de commandes (l'interface) entre l'utilisateur et le système d'exploitation, d'où son nom anglais «shell», qui signifie «coquille».

Le shell est ainsi chargé de faire l'intermédiaire entre le système d'exploitation et l'utilisateur grâce aux lignes de commandes saisies par ce dernier. Son rôle consiste ainsi à lire la ligne de commande, interpréter sa signification, exécuter la commande, puis retourner le résultat sur les sorties.

Le shell affiche une invite en début de ligne, appelée prompt ('\$' ou '#' ou '%'), pour indiquer à l'utilisateur qu'il attend l'entrée d'une commande

### 1.2 Pourquoi utiliser le mode console (= mode terminal virtuel) ?

Le mode console ressemble visuellement au mode ms-dos de Microsoft (mode texte); Il s'agit toutefois d'un Linux complet en mode multi-taches et multi-utilisateur; Ce mode est utile dans 2 cas précis :

- on désire disposer d'un maximum de puissance (le système graphique est consommateur en ressources processeur et mémoire)
- Le système graphique n'est pas disponible pour diverses raisons (panne, terminal en mode texte..)
- contrairement à ce que l'on pourrait croire, le mode console est plus simple que le mode graphique pour certaines opérations (ex : montage de lecteur, transfert d'un fichier, ajout d'un utilisateur,...)

### 1.3 Que peut-on faire en mode console ?

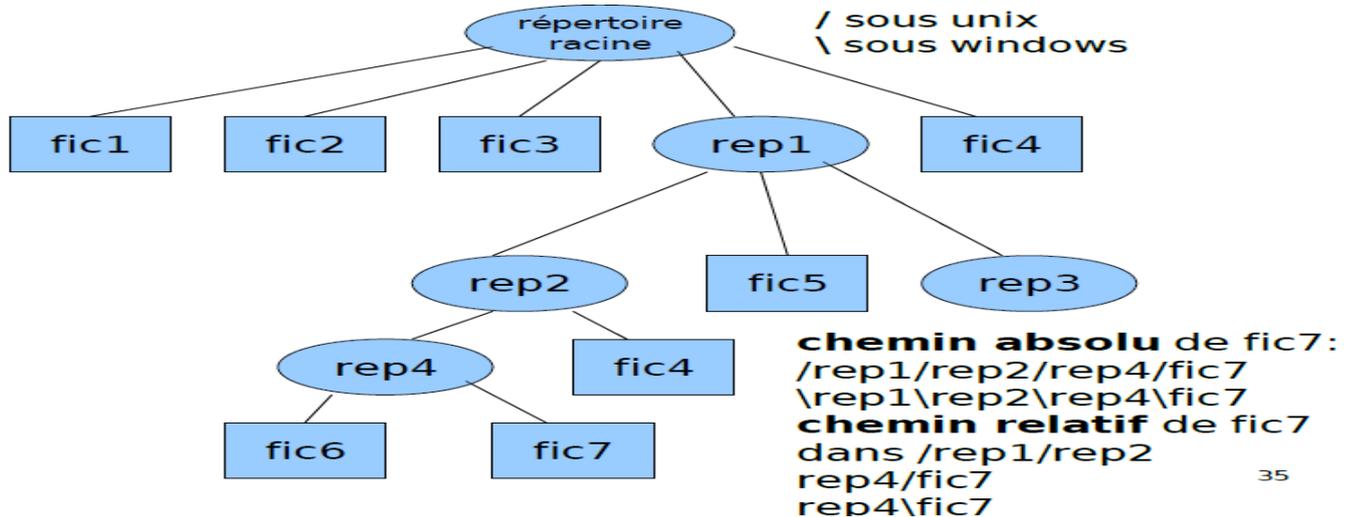
A peu près tout (logiciels en ligne de commande) ce qu'on peut faire en mode graphique, à l'exception évidemment des applications graphiques (dessin, photos, vidéos, ...)

## 2 Consulter le manuel d'une commande

*man* *commande*

Affiche la page du manuel correspondant à la commande spécifiée.

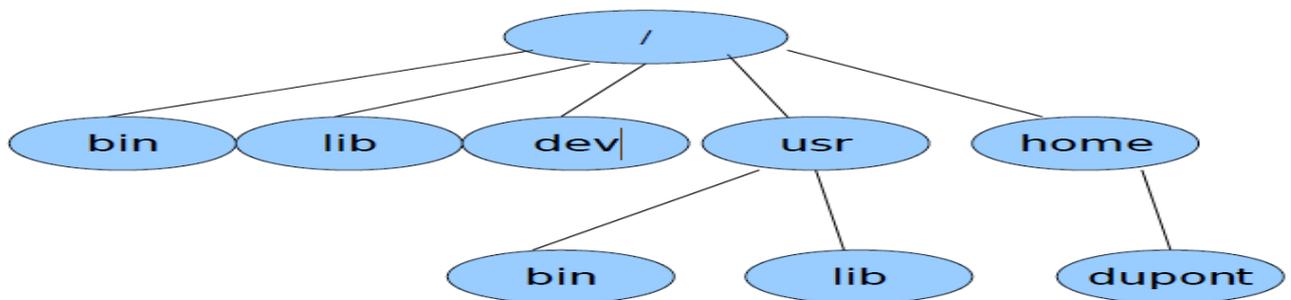
## 3 Commandes et manipulation des répertoires



35

**Figure : arborescence des fichiers**

### arborescence classique sous unix



### 3.1 Visualiser le contenu d'un répertoire

*Ls* : Affiche (liste) le contenu d'un répertoire (répertoire courant si non spécifié).

### 3.2 Changer un répertoire

*cd chemin* : change le répertoire courant par celui spécifié par le chemin.

*cd /* : change le répertoire courant par le home directory.

*cd ..* : change le répertoire courant par le répertoire précédent

*pwd* : (print working directory) affiche le chemin du répertoire courant

### 3.3 Créer et détruire un répertoire

*mkdir* *nom\_répertoire* : Création d'un répertoire portant le nom « nom\_répertoire ».

**rmdir** *nom\_répertoire* : Supprime un répertoire vide (pour supprimer un répertoire non vide, il faut utiliser la commande *rm*)

## 4 Commandes et manipulation des fichiers

### 4.1 Visualiser le contenu d'un fichier

**cat** [*fichiers*] : Cette commande permet d'afficher un ou plusieurs fichiers sur la sortie standard.

### 4.2 Chercher une chaîne de caractères dans un fichier

**grep** [*-option*] *expression* *fichiers*

sans option : recherche dans les fichiers les lignes contenant l'expression

**-i** : pour ne pas tenir compte des majuscules/minuscules

**-v** : pour afficher les lignes ne contenant pas l'expression spécifiée.

### 4.3 Faire une copie d'un fichier

**cp** *source* *destination*

Copie le fichier source dans le fichier destination.

Si le fichier destination n'existe pas, il est créé. Sinon son contenu est écrasé sans avertissement.

Si la destination est un répertoire, alors la source peut être une liste de fichiers.

### 4.4 Déplacer ou changer le nom d'un fichier

**mv** *source* *destination*

Renomme ou déplace le fichier source en destination. Si la destination est un répertoire, alors la source peut être une liste de fichiers.

### 4.5 Détruire un fichier

**rm** [*-option*] *fichiers* : Efface les fichiers (attention, on ne peut pas récupérer un fichier qui a été effacé), on a 3 options de suppression :

**-i** : Interactif, demande une confirmation de suppression pour chaque fichier

**-f** : force la suppression du fichier

**-r** : Récursivité, permet d'effacer un répertoire et son contenu.

### 4.6 Permission sur les fichiers

Pour chaque fichier, il y a trois classes d'utilisateurs

utilisateur : le propriétaire du fichier

groupe : le groupe auquel appartient le fichier

autre : tous les autres

Les permissions accordées à ces trois classes sont :

**r** : lecture ; **w** : écriture - **x** : exécution

**chmod** *mode* *fichiers*

change les permissions du ou des fichiers/répertoires.

*exemple* mode désiré : rwxr-xr--

<i>user group other</i>		
rwx	r-x	r--
111	101	100
7	5	4

en binaire

d'ou la commande **chmod 754 fichier**

### **Symboles spéciaux dans les chemins:**

- ✓ `..` : désigne le répertoire parent du répertoire courant.
- ✓ `~` : désigne votre répertoire personnel, `/home/dupont` par exemple.
- ✓ `.` : désigne le répertoire courant.

## **5 Gestion des utilisateurs**

### **5.1 Identifier les utilisateurs du système**

**who** : Fournit des informations sur l'ensemble des utilisateurs qui sont actuellement connectés sur la station (machine).

**id** : renvoie l'UID (user identifier), le GID (Groupe identifier) de l'utilisateur courant.

### **5.2 Changer un mot de passe**

Pour se connecter, il faut :

- un login (identificateur de l'utilisateur) assigné par votre administrateur système
- un password (mot de passe) propre à chaque utilisateur

#### **passwd**

Permet de définir et de contrôler son mot de passe.

A l'appel de cette commande, vous devez saisir l'ancien mot de passe, puis vous devez saisir deux fois votre nouveau mot de passe.

### **5.3 Se connecter sur un autre compte**

**su** *[-]* *utilisateur*

Change l'utilisateur courant. Vous devez saisir le mot de passe du nouvel utilisateur (sauf si le compte d'origine est le root).

Si vous indiquez - alors les fichiers de login sont exécutés et vous vous retrouvez dans le home directory du nouveau compte.

### **5.4 Ajouter un utilisateur**

Passer en session de travail **root** Il est tout-à-fait déconseillé de se connecter comme *root* pour une session de travail, le risque d'endommager le système est trop grand, même si on le connaît bien !.

Pourtant, certaines commandes ou accès à des répertoires nécessitent d'avoir les droits de *root*, le compte administrateur du système. En pratique, on se connecte comme utilisateur normal et on ne "devient" root qu'en cas de nécessité. Et on quitte la session root aussitôt après. Pour ajouter un utilisateur, basculer vers le compte du Root avec la commande : **su**

ensuite introduisez le Password du Root ; ensuite ajouter un User avec la commande : **adduser**

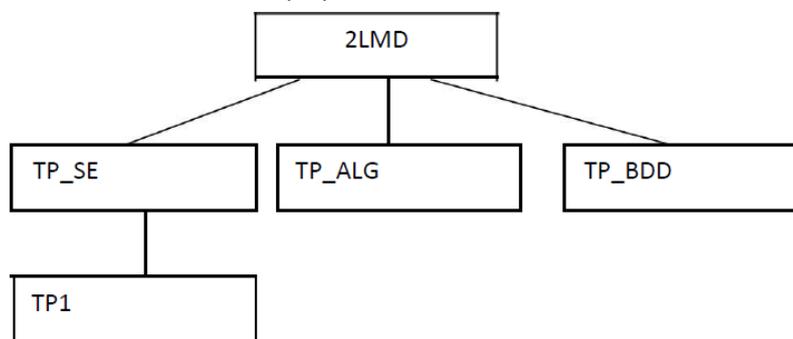
```
# adduser user1
```

Vérifiez en vous connectant comme user1 dans une autre console

## Application

### Exercice 1 :

1. Donnez une suite de commandes qui permet la création de l'arborescence suivante (dans le répertoire home):



2. Ajoutez les commandes qui permettent de:

- Listez le contenu du répertoire 2LMD
- Renommez le répertoire TP\_ALG en TP\_POO
- Se déplacez vers TP1
- Allez vers le répertoire parent
- Supprimez le répertoire TP\_SE

3. Quelle est la commande qui permet de connaître le chemin du répertoire où l'on se trouve (nom absolu)?

4. Où doit-on se placer et exécutez la commande de la question 3 pour avoir le chemin suivant :

**/home/2LMD/TP\_BDD**

5. Supprimez le répertoire TP\_BDD

6. Créez deux répertoires TP1 et TP2 dans 2LMD

7. Déplacez TP1 dans TP\_POO

8. Copiez TP2 dans TP1

Proposé par l'enseignant : Omar BOUKADOUM. E\_mail : [BOUKADOUM2020@gmail.com](mailto:BOUKADOUM2020@gmail.com). Facebook : [www.facebook.com/boukadoumomar](http://www.facebook.com/boukadoumomar)