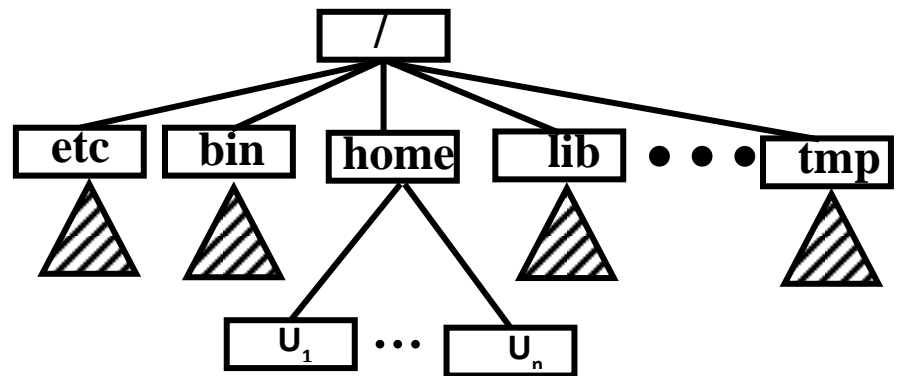


Les fichiers répertoires

Parmi les qualités d'un système d'exploitation, on trouve le système de fichiers. En effet, celui-ci constitue la partie la plus évidente dans un système d'exploitation.

Dans le système LINUX (Windows), la gestion de fichiers est basée sur une organisation hiérarchique de fichiers répertoires (catalogues ou dossiers) qui contiennent un ensemble fini de fichiers. Chaque fichier répertoire est lui aussi contenu dans un autre. Cette structure hiérarchique se présente sous forme d'**arborescence** dont la racine (**root**) est représenté par le caractère "/" ou par "\" pour le système Windows.

Un catalogue (ou répertoire ou dossier) est un fichier qui peut être défini comme un point d'entrée particulier dans le système de fichier (S.F.) et à partir du quel on peut créer n'importe quel type de fichier y compris des fichiers répertoires.



Exemple d'arborescence

1.1) Les fichiers répertoires standards

Le système **LINUX** possède un certain nombre de répertoires standards :

- /etc** : Les commandes et les fichiers de l'administrateur système.
- /bin** : Les commandes principales de l'utilisateur.
- /lib** : Les bibliothèques importantes.
- /tmp** : Les fichiers temporaires.
- /home** : Les différents utilisateurs du système.

1.2) Notion de chemin

Syntaxe : `pwd [options]`

Quel que soit la position courante dans l'arborescence, la commande "**pwd**" (print **w**orking **d**irectory) nous indique le chemin entre la racine (/) et le répertoire courant. Les différents sommets ou nœuds qui appartient à ce chemin sont séparés par le caractère "/".

Exemple : Après connexion, la commande "pwd" donne : **/** home / u₁

- Le premier **" / "** est la racine de l'arborescence.
- **" u₁ "** est le nom de l'utilisateur.
- **home** est le répertoire qui contient les noms de tous les utilisateurs.
- Les autres **" / "** excepté le premier sont des **séparateurs** de nœuds.

Remarques : Une **référence** est dite **absolue** si elle commence par **" / "**. Par convention, une référence est dite **relative** si elle ne commence pas par **" / "**.

1.3) Création d'un fichier répertoire

Syntaxe : `mkdir` □ [options] [chemin/] nom_rep

- **mkdir** (**make directory**) est le nom de la commande.
- "nom_rep" désigne le nom du fichier répertoire à créer. Ce nom est choisi par l'utilisateur.
- "chemin" est facultatif. Il s'agit d'une référence absolue ou relative.

Effet : Si "chemin" est erroné ou "nom_rep" existe alors une erreur est affichée. Sinon "nom_rep" est créé sous le répertoire indiqué par la référence "chemin" ou sous le répertoire courant si "chemin" est omis. Après création du fichier répertoire "nom_rep", celui-ci contiendra deux sous répertoires cachés nommés "." et "..". Le "." indique le répertoire courant (nom_rep) et le ".." indique le répertoire père du répertoire "nom_rep".

Remarques : Un fichier répertoire est dit **vide** si et seulement s'il contient uniquement les deux fichiers répertoires cachés : "." et "..".

1.4) Destruction d'un fichier répertoire

Syntaxe : `rmdir` □ [options] [chemin/] nom_rep

- **rmdir** (**remove directory**) est le nom de la commande.
- "nom_rep" désigne le nom du fichier répertoire à détruire.
- "chemin" est facultatif. Il s'agit d'une référence absolue ou relative.

Effet : Si "chemin" est erroné ou "nom_rep" n'est pas un fichier répertoire **vide** alors une erreur est affichée. Sinon "nom_rep" est détruit y compris ses deux sous fichiers répertoires : "." et "..".

Remarque : Pour détruire un dossier non vide, il faut utiliser la commande **rm** :

Syntaxe : `rm` □ -r □ [chemin/] nom_rep

Cette commande détruit le dossier nom_rep ainsi que son contenu.

1.5) Contenu d'un fichier répertoire

Syntaxe : `ls` □ [options] [options] [chemin]

- **ls** (liste contents directory) est le nom de la commande.
- "chemin" est facultatif. Il s'agit d'une référence absolue ou relative.
- options sont indiquées dans le résumé de la commande "ls".

Effet : Si "chemin" est erroné alors une erreur est affichée. Sinon la liste de fichiers qui se trouve dans le répertoire spécifié par "chemin" est affichée. Si "chemin" est omis alors le contenu du répertoire courant est affiché.

Ci-dessous, le résultat de `ls -li`

```
mil@debian:~$ ls -li
total 48
131127 drwxrwxrwx 5 mil mil 4096 mars 4 2020 additions
131099 drwxr-xr-x 4 mil mil 4096 août 29 08:29 Bureau
131268 drwxr-xr-x 2 mil mil 4096 sept. 3 15:09 cmd
131103 drwxr-xr-x 2 mil mil 4096 mars 4 2020 Documents
131105 drwxr-xr-x 2 mil mil 4096 mars 4 2020 Images
131101 drwxr-xr-x 2 mil mil 4096 mars 4 2020 Modèles
131104 drwxr-xr-x 2 mil mil 4096 mars 4 2020 Musique
131102 drwxr-xr-x 3 mil mil 4096 mars 6 13:25 Public
131203 drwxr-xr-x 2 mil mil 4096 sept. 2 07:17 s1
131244 drwxrwx--- 8 mil mil 4096 août 31 17:14 s220
132594 -rw----- 1 mil mil 0 août 31 10:22 t
131248 lrwxrwxrwx 1 mil mil 1 sept. 4 09:19 tbis -> t
131100 drwxr-xr-x 2 mil mil 4096 août 31 17:11 Téléchargements
131106 drwxr-xr-x 2 mil mil 4096 mars 4 2020 Vidéos
```

-
- ① Numéro de nœud du fichier
 - ② Type de fichier
 - ③ Mode de fichier
 - ④ Nombre de liens
 - ⑤ Nom du propriétaire
 - ⑥ Nom du groupe
 - ⑦ Taille du fichier (nombre d'octets)
 - ⑧ Date de création
 - ⑨ Nom du fichier

Remarques : Plusieurs types existent. Voici quelques exemples

Mon du type de fichier	Caractère utilisé par ls
Ordinaire ou normal	-
Répertoire ou dossier	d
Lien symbolique	l
Spécial bloc	b
Spécial caractère	c

1.6) Changement d'un fichier répertoire

Syntaxe : `cd` □ [chemin/] nom_rep

- `cd` (change working directory) est le nom de la commande.
- "nom_rep" désigne le nom du fichier répertoire à accéder.
- "chemin" est facultatif. Il s'agit d'une référence absolue ou relative.

Effet : D'une façon générale si "chemin" est erroné ou "nom_rep" n'est pas un fichier répertoire alors une erreur est affichée. Sinon on réalise un accès au fichier répertoire nommé "nom_rep".

Exemples :

```
cd □ rep    # Réalise un accès au dossier rep (on suppose l'existence de ce dossier)
cd □ .      # On est toujours sous le dossier rep, car "." désigne aussi le dossier rep.
cd □ ..     # Réalise un accès au dossier père de rep ( ".." désigne aussi le dossier père de rep)
```

Remarques : Quel que soit la position courante d'un utilisateur, la commande `cd` sans arguments remplace cet utilisateur sous son répertoire de connexion.

1.7) Exploration de l'arborescence

Explorer l'arborescence dans le sens descendant :

Voir ci-contre, figure n°1 : Vous êtes sous le répertoire "leo".

Comment se déplacer pour aller s'installer sous le répertoire "rep3" ?

Réponse : il faut explorer l'arborescence dans le sens **descendant**

Il faut passer par les répertoires : "tp1" et "rep3".

De plus, il faut séparer ces derniers par "/".

La commande est : **cd tp1/rep3**

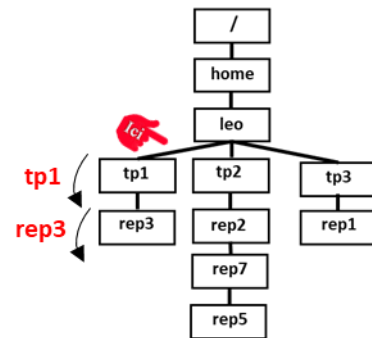


Figure n°1

Explorer l'arborescence dans le sens ascendant :

Voir ci-contre, figure n°2 : Vous êtes sous le répertoire "rep1".

Comment se déplacer pour aller s'installer sous le répertoire "leo" ?

Réponse : Il faut **rebrousser** son chemin vers le père de "rep1". Ici le père de "rep1" est : "..".

Sous "rep3", Il faut de nouveau **rebrousser** son chemin vers le père de "rep3".

Ici le père de "rep3" est : "..".

De plus, il faut séparer les deux répertoires par "/".

La commande est : **cd ../..**

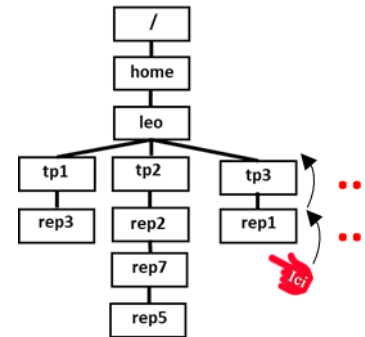


Figure n°2

Construction de chemin Absolu et relatif :

Voir ci- contre, figure n°3 : Vous êtes sous le répertoire "rep1".

Donner le chemin **absolu** et **relatif** qui mène vers "rep5"

Réponses :

- Un chemin **absolu** commence toujours par **/**.

Ce chemin est : **/home/leo/tp2/rep2/rep7**

- Le chemin **relatif** par rapport à "rep1" est le suivant :

Il faut rebrousser son chemin deux fois pour être sous le répertoire "leo".

De plus, il faut passer par les répertoires suivants : "tp2", "rep2" et "rep7"

Donc le chemin relatif est : **../.. / tp2/rep2/rep7**

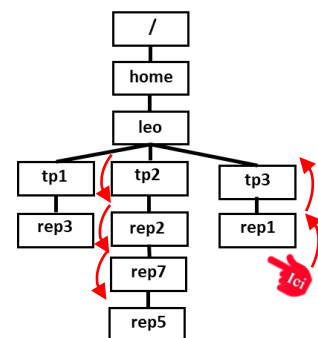


Figure n°3

1.8) Abréviations de noms de fichiers répertoires

Le tableau ci-dessous donne des noms et des abréviations pour manipuler des fichiers répertoires :

Nom du fichier	Commentaire
.	Répertoire courant.
..	Répertoire père du répertoire courant.
~	Référence absolue du répertoire de connexion.
~nom_u	Si "nom_u" est le nom de l'utilisateur alors ~nom_u est idem à ~. Sinon une erreur est affichée.
~-	La référence du répertoire précédant.
~+	La référence du répertoire courant.

1.9) Variables shell pour les fichiers répertoires

La liste ci-dessous donne le nom de quelques variables utiles pour manipuler les fichiers répertoires. Pour obtenir la valeur de la variable, il faut précéder le nom de la variable par le caractère "\$" (Exemple : **HOME** est le nom de la variable et **\${HOME}** est la valeur de cette variable).

HOME : Contient la référence absolue du fichier répertoire de connexion.
PWD : Contient la référence absolue du fichier répertoire courant.
OLDPWD : Contient la référence absolue du fichier répertoire précédant.
CDPATH : Contient une liste de répertoires.
 Les éléments de la liste sont séparés par le caractère ":".

Lors de l'exécution de la commande `cd` □ nom_rep :

Si "nom_rep" existe sous le répertoire courant alors l'accès au dossier "nom_rep" se réalise avec succès. Sinon on cherche "nom_rep" parmi les éléments de la variable CDPATH. Si le résultat de la recherche abouti alors l'accès au dossier "nom_rep" se réalise avec succès. Sinon une erreur sera affichée.

Exemple : Au niveau système, on réalise l'affectation suivante : **CDPATH = : : ~/rep**

Dans cet exemple la variable CDPATH contient deux répertoires : le répertoire courant, la référence absolue du fichier répertoire "rep" (on suppose que le répertoire "rep" existe sous le répertoire de connexion). L'exécution de la commande `cd` □ **nom_rep** se déroulera de la façon suivante : le système essaye de réaliser un accès au répertoire nommé "mon_rep" d'abord sous le répertoire courant, puis, dans le cas d'un échec, le système essaye un deuxième accès au répertoire "mon_rep" mais dans : ~/rep.

Remarques : Les écritures suivantes sont identiques :

`cd` □ ~ est identique à `cd` **\${HOME}**
`cd` □ ~- est identique à `cd` **\${OLDPWD}**
`cd` □ ~+ est identique à `cd` **\${PWD}**

1.10) Commandes sur les fichiers répertoires

Voir le résumé de commandes sur les fichiers répertoires et tester le maximum d'options

- **pwd**
- **cd**
- **mkdir**
- **rmdir**
- **ls**