

Se repérer dans l'arborescence

Système de fichiers, chemins absolus et chemins relatifs

Objectifs de ce chapitre

- Expliquer ce qu'est un système de fichiers et une arborescence.
- Se repérer dans l'arborescence : racine, répertoire personnel, répertoire courant.
- Distinguer et écrire un chemin absolu et un chemin relatif.
- Se déplacer avec les commandes `pwd`, `ls` et `cd`.

Où sont mes fichiers ?

Sur un téléphone ou une tablette, les applications rangent tes données à ta place : tu ouvres la galerie pour voir tes photos sans jamais savoir où elles sont stockées. Sur un ordinateur, il faut savoir où ranger ses fichiers et comment les retrouver. Le système d'exploitation t'aide pour cela grâce à son **système de fichiers**.

Définition — système de fichiers

Le **système de fichiers** est la façon dont le système d'exploitation organise et retrouve les données sur un support de stockage (disque, SSD, clé USB). Pour l'utilisateur, il se présente sous la forme d'une **arborescence** : des **répertoires** (les dossiers) qui contiennent des **fichiers** et d'autres répertoires.

Remarque : l'expression désigne aussi la façon dont les données sont écrites physiquement sur le support (ext4, NTFS, exFAT...). Ici, seule la vue de l'utilisateur nous intéresse.

Activité 1 — explore avec le gestionnaire de fichiers

Ouvre le gestionnaire de fichiers de ton poste (l'icône en forme de dossier) et réponds aux questions.

1. Dans quel dossier te trouves-tu à l'ouverture ? Il s'agit de ton dossier personnel.
2. Affiche la barre d'adresse avec **Ctrl+L**. Quel chemin est écrit ? Note-le.
3. Ouvre le dossier Documents. Comment le chemin change-t-il ?
4. Remonte d'un niveau à la fois (flèche vers le haut, ou **Alt+↑**) jusqu'à ne plus pouvoir. Quel est le dernier dossier atteint ? Comment est-il noté ?
5. Dans ce dossier, repère les noms `home`, `etc`, `usr`. Où se trouve ton dossier personnel ?
6. (*Démonstration ou si possible*) Branche une clé USB. Où apparaît-elle dans l'arborescence ?

Ce que tu viens de découvrir

- Il n'y a **qu'une seule arborescence**, qui part d'un point de départ unique : la **racine**, notée `/`.
- Contrairement à Windows (`C:`, `D:...`), une clé USB n'a pas de « lettre » : elle apparaît à l'**intérieur** de l'arborescence, en général sous `/media`.
- Le chemin affiché dans la barre d'adresse décrit où l'on se trouve : c'est exactement ce que nous allons apprendre à écrire.

I. L'arborescence

Les fichiers sont organisés comme un arbre inversé, dont la racine `/` contient tous les autres répertoires et fichiers. Voici un exemple, que nous réutiliserons dans ce chapitre :

```

/
|-- boot
|   |-- grub
|   |   |-- grub.cfg
|-- etc
|   |-- passwd
|-- home
|   |-- elsa
|   |   |-- documents
|   |   |   |-- fiche.ods
|   |   |   |-- rapport.odt
|   |   |-- musique
|   |   |   |-- morceau.mp3
|   |-- max
|   |   |-- boulot
|   |   |-- images
|   |   |   |-- photo_1.jpg
|-- usr
|   |-- bin

```

Dans cet exemple, `grub.cfg`, `passwd`, `fiche.ods`, `rapport.odt`, `morceau.mp3` et `photo_1.jpg` sont des fichiers. Tous les autres noms sont des répertoires.

1. Le vocabulaire de base

Terme	Signification
fichier	Un ensemble de données portant un nom : document, image, programme...
répertoire (ou dossier)	Un conteneur qui peut contenir des fichiers et d'autres répertoires.
racine	Le répertoire de départ de toute l'arborescence, noté <code>/</code> .
répertoire parent	Le répertoire qui contient un autre répertoire. Le parent de <code>/home/elsa</code> est <code>/home</code> .
répertoire personnel	Ton espace de travail, par exemple <code>/home/elsa</code> . Il se note aussi <code>~</code> .
répertoire courant	L'endroit où l'on se trouve à un instant donné dans l'arborescence. Il se note <code>.</code> .
chemin	La suite de répertoires à parcourir pour atteindre un fichier ou un répertoire.

⚠ Les majuscules comptent

Les systèmes de type Unix sont **sensibles à la casse** : le répertoire `Documents` n'est pas le répertoire `documents`. Choisis aussi des noms sans espace et sans accent, ce qui évite bien des problèmes.

2. Les répertoires fondamentaux

Quelques répertoires présents à la racine de tout système Linux, et leur rôle :

Répertoire	Contenu
<code>/home</code>	les répertoires personnels des utilisateurs (par exemple <code>/home/elsa</code>)
<code>/etc</code>	les fichiers de configuration du système

Répertoire	Contenu
/bin	les commandes essentielles (ls, cp...)
/usr	les programmes et données partagés par tous les utilisateurs
/var	les données qui changent : journaux (logs), caches...
/tmp	les fichiers temporaires
/dev	des fichiers qui représentent les périphériques (disque, terminal...)
/media	les supports amovibles, comme les clés USB
/boot	le noyau et les fichiers nécessaires au démarrage

II. Le terminal : où suis-je ?

Le **terminal** est la fenêtre dans laquelle on tape des commandes. Elles sont lues par le **shell**, qui fait l'interface entre l'utilisateur et le noyau. Sous Linux, le shell le plus courant s'appelle **Bash**.

À l'ouverture, le terminal affiche une **invite de commande** (le *prompt*) qui ressemble à ceci :

```
prenom@poste2:~$
```

Élément	Signification
prenom	le nom de l'utilisateur sur cette machine
poste2	le nom de la machine
~	le répertoire courant : ici ton répertoire personnel, c'est-à-dire /home/prenom
\$	l'invite d'un utilisateur ordinaire (pour l'administrateur, c'est #)

Remarque : le caractère `~` s'obtient avec `AltGr+2` puis un appui sur la barre d'espace.

1. Les commandes pour se déplacer

Commande	Effet	Exemple
pwd	Affiche le chemin absolu du répertoire courant (<i>Print Working Directory</i>).	pwd
ls	Liste le contenu du répertoire courant (<i>LiSt</i>).	ls
ls chemin	Liste le contenu du répertoire indiqué, sans s'y déplacer.	ls /home
ls -a	Liste aussi les fichiers cachés, dont le nom commence par un point.	ls -a
cd chemin	Change de répertoire courant (<i>Change Directory</i>).	cd /home/elisa
cd ..	Remonte dans le répertoire parent.	cd ..
cd ou cd ~	Retourne dans le répertoire personnel.	cd
cd -	Retourne dans le répertoire précédent.	cd -
man commande	Affiche le manuel d'une commande (touche <code>q</code> pour quitter).	man ls

Remarque : un espace sépare la commande de ses paramètres, et les options commencent par un tiret.

**Deux astuces qui changent la vie**

- Appuyez sur **Tab** pour compléter automatiquement le nom d'un fichier ou d'un répertoire (deux appuis affichent les possibilités).
- Appuyez sur **↑** pour rappeler les commandes précédentes, sans les retaper.

**Activité 2 — déplace-toi au terminal**

Ouvrez un terminal et répondez aux questions. Adaptez les noms si vos dossiers s'appellent autrement.

1. Lisez le prompt : repérez le nom d'utilisateur, le nom de la machine et le répertoire courant.
2. Tapez `pwd`. Comparez avec le chemin noté lors de l'activité 1.
3. Tapez `ls`, puis `ls -a`. Qu'y a-t-il en plus ? Repérez en particulier `.` et `..`.
4. Tapez `cd Documents`, puis `pwd`. Que devient le prompt ?
5. Tapez `cd ..`, puis `pwd`. Où êtes-vous ?
6. Tapez `cd /`, puis `ls`. Retrouvez-vous ce que vous avez vu dans le gestionnaire de fichiers ?
7. Tapez `cd` sans rien d'autre, puis `pwd`. Où êtes-vous revenu ?
8. Tapez `cd documents` (avec un `d` minuscule). Que se passe-t-il ? Pourquoi ?

III. Chemins absolus et chemins relatifs

Pour désigner un fichier ou un répertoire, il faut écrire son **chemin**. Il y a deux façons de le faire. Les exemples utilisent l'arborescence de la partie I.

**Définition — chemin absolu**

Un **chemin absolu** commence toujours par `/` et décrit le trajet depuis la racine. Il ne dépend pas de l'endroit où l'on se trouve.

Exemple : `/home/elsa/documents/fiche.ods`

Remarque : seul le premier `/` désigne la racine, les suivants sont de simples séparateurs.

**Définition — chemin relatif**

Un **chemin relatif** ne commence pas par `/` : il décrit le trajet depuis le **répertoire courant**. Il dépend donc de l'endroit où l'on se trouve.

- `.` désigne le répertoire courant (on peut souvent l'omettre) ;
- `..` désigne le répertoire parent, c'est-à-dire le répertoire du dessus.

Exemples : on se trouve dans `/home/elsa`

Pour atteindre...	Chemin relatif	Chemin absolu
le fichier <code>fiche.ods</code>	<code>documents/fiche.ods</code>	<code>/home/elsa/documents/fiche.ods</code>
le répertoire <code>images</code>	<code>../max/images</code>	<code>/home/max/images</code>
le répertoire <code>home</code>	<code>..</code>	<code>/home</code>
la racine	<code>../..</code>	<code>/</code>

Méthode pour écrire un chemin relatif

1. Repérez le point de départ (le répertoire courant) et la cible.
2. **Remontez** avec `..` jusqu'au premier répertoire qui contient à la fois le départ et la cible.
3. **Redescendez** ensuite vers la cible en écrivant les noms des répertoires.

Exemple : de `/home/elsa` vers `/home/max/images`. Le premier répertoire commun est `/home` : on remonte d'un niveau (`..`), puis on redescend par `max/images`. Le chemin est `../max/images`.

Un chemin relatif dépend du point de départ

Le chemin relatif `../max` désigne `/home/max` depuis `/home/elsa`, mais il désigne `/home/max/max` (qui n'existe pas) depuis `/home/max/images`. Avant d'écrire un chemin relatif, demandez-vous toujours : « où suis-je ? »

À retenir

L'essentiel

- Le **système de fichiers** organise les données en une **arborescence** unique, qui part de la racine `/`.
- `pwd` donne le répertoire courant, `ls` liste son contenu, `cd` change de répertoire.
- Un **chemin absolu** commence par `/` et ne dépend pas du répertoire courant.
- Un **chemin relatif** part du répertoire courant : `.` est le répertoire courant, `..` son parent.
- Les noms sont **sensibles à la casse** : `Documents` n'est pas `documents`.

S'entraîner avec des jeux

Deux jeux permettent de s'exercer à se déplacer dans une arborescence, jusqu'à être à l'aise avec les chemins absolus et relatifs :

- [Jeu FYP \(université Lyon 1\)](#)
- [Terminus](#), la traduction française du jeu du MIT. Utilisez la fiche d'aide distribuée par ton professeur.

À toi de jouer

Pendant le jeu, note dans votre cahier trois déplacements écrits avec un **chemin relatif**, et leur équivalent avec un **chemin absolu**.

Exercices

Exercice 1 — sur papier, avec l'arborescence de la partie I

Écrivez le chemin demandé.

- Chemin absolu du fichier `grub.cfg` :
- Chemin absolu du répertoire `boulot` :
- Chemin relatif de `rapport.odt` depuis `/home/elsa/documents` :
- Chemin relatif du répertoire `boulot` depuis `/home/elsa/documents` :
- Chemin relatif de `photo_1.jpg` depuis `/home/elsa/musique` :
- Chemin relatif de `morceau.mp3` depuis `/etc` :
- Chemin relatif de `passwd` depuis `/usr/bin` :

Exercice 2 — la chasse aux chemins (sur machine)

Votre but : écrire des chemins vers des répertoires que tu peux lire, puis vérifier vous-même qu'ils sont justes.

Partie A. Placez-vous dans ton répertoire personnel avec `cd`, puis notez le résultat de `pwd` :

Pour chaque cible du tableau, écrivez d'abord ses deux chemins, **sans les tester**. Vérifiez ensuite :

- `ls chemin` : si le shell répond « Aucun fichier ou dossier de ce type », ton chemin est faux ;
- `realpath chemin_relatif` : affichez le chemin absolu qui correspond à ton chemin relatif. Comparez-le à votre réponse.

Cible	Chemin absolu	Chemin relatif depuis mon répertoire courant	Vérifié
<code>/etc</code>			
<code>/usr/share/doc</code>			
<code>/tmp</code>			
<code>/var/log</code>			
<code>/home</code>			

Remarque : le nombre de `..` dépend de la place de ton répertoire personnel dans l'arborescence. Regarde le résultat de `pwd` pour le déterminer.

Exercice 2 (suite)

Partie B. Changez de point de départ : tape `cd /usr/share`. Écris, puis vérifiez, le chemin relatif vers `/etc`, vers `/tmp` et vers votre répertoire personnel.

Partie C. Tapez `ls /root`. Que se passe-t-il ? Notez le message. On l'expliquera dans le chapitre sur les droits.

Défi. Depuis `/usr/share`, retrouvez votre répertoire personnel d'au moins trois façons différentes :

.....