

Mémo ligne de commande

BARBIER J.M. - LGT Dumezil - 1NSI

12 mars 2026

1 Invite de commande / prompt

1.1 Lecture du prompt

utilisateur@nomduserveur:emplacement_ds_arborescence\$...

~ = répertoire HOME (souvent /home/*nomutilisateur*/)

2 Naviguer dans l'arborescence

2.1 Exemple d'arborescence

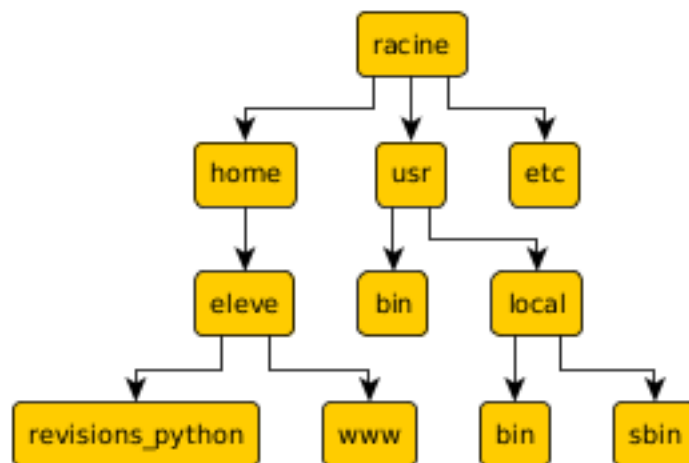


Figure 1 - Arborescence fichiers

2.2 Naviguer

- pour savoir où on est : `pwd` (renvoie le chemin complet du répertoire courant)
- pour changer de répertoire : `cd` (change directory)
 - `cd` = retour à la maison (HOME)
 - `cd ..` = aller au répertoire parent
 - `cd nomdurépertoire` = aller dans le répertoire indiqué en argument

Le nom d'un répertoire peut être donné de plusieurs manières :

- de manière **absolue** : à partir de la racine, exemple `/usr/local/bin`
- de manière **relative** : à partir du répertoire où on est `../usr/local/bin` ou `dossier/ssdossier`

2.3 Lister le contenu

- pour lister le contenu d'un répertoire : `ls` (répertoire courant) ou `ls nom_fichier_ou_dossier`
 - `ls -l` = affiche le détail du contenu
 - `ls -al` = affiche le détail + les fichiers / dossiers cachés (commence par `.`)
 - `ls -alh` = pareil, mais avec des unités "humaines"

2.4 Pour être plus efficaces

- la touche TAB complète la commande, en une fois si aucune ambiguïté
- si ambiguïté, le deuxième appui sur TAB indique les choix possibles compatibles avec ce qui a déjà été tapé
- la flèche vers le haut rappelle les commandes déjà tapées (historique)
- `Ctrl+R` cherche dans l'historique, chaque `ctrl+R` remonte d'un cran

3 Gestion des fichiers

3.1 Créer / supprimer des fichiers et répertoires

- `touch nom_du_fichier` = crée un fichier vide
- `mkdir nom_du_repertoire` = créer un répertoire
- `rm nom_du_fichier` = ReMove supprime un fichier

Pour supprimer un répertoire :

- `rmdir nom_du_repertoire` supprime le répertoire si il est vide
- `rm -rf nom_du_repertoire` supprime le répertoire et tout son contenu (**DANGEREUX**)

Renommer un fichier / déplacer un fichier :

- `mv ancien_nom nouveau_nom` : renomme le fichier de ancien nom vers nouveau nom
- `mv nom_fichier nom_repertoire/` : déplace le fichier (sans le renommer) vers le répertoire

3.2 Lire / gérer les permissions

Savoir lire toutes les informations d'un fichier : `ls -l nom_du_fichier`

- 3 niveaux d'autorisations
 - pour le propriétaire (u)
 - pour un groupe (g)
 - pour le reste du monde (o)
- 3 permissions
 - lecture (r)
 - écriture (w) : pour fichier = modifier / pour répertoire = créer fichier dedans
 - exécution (x) : pour fichier = lancer le programme / pour répertoire = aller dedans

```
uuugggooo    prop  gpe
-rw-r--r-- 1 eleve eleve 1385 sept. 14 16:59 ligne-commande.md
```

Notation des permissions en octal : chaque permission sur 3 bits

exemples :

```
rw----- = 111 000 000 = 700
r-xr-xr-- = 101 101 100 = 554
rw-rw-r-- = 110 110 100 = 664
331       = 011 011 001 = -wx-wx--x
753       = 111 101 011 = rwxr-x-wx
```

Exercice :

- Je veux faire un fichier lisible et exécutable par moi et par le groupe, mais pas les autres = `r-xr-x---` = 550
- Je veux faire un fichier lisible par moi et le groupe, où moi et le groupe pouvons écrire dedans, et rien les autres = `rw-rw----` = 660

Pour changer les droits d'un fichier : `chmod XXX nomdufichier` (le propriétaire du fichier peut toujours changer les droits du fichier, même si il s'est enlevé tous les droits dessus)

3.3 Consulter et modifier un fichier

Pour afficher le contenu d'un fichier :

- `cat nomdufichier` = affiche le contenu du fichier
- `more nomdufichier` = affiche le contenu du fichier avec une pagination

Pour modifier un fichier : `vi nomdufichier` (présent sur à peu près 100% des machines)

- passer en mode insertion :  puis 
- enregistrer :  puis  puis 
- pour quitter :
 - pour quitter en enregistrant :  puis  puis  puis 
 - pour quitter sans enregistrer :  puis  puis  puis 

Les autres possibles : nano / emacs (selon les systèmes)

4 Autres commandes utiles

4.1 Documentation

- `man commande` = affiche page de manuel de la commande
- `apropos mot` = liste toutes les pages de manuel qui contiennent ce mot

En passant : RTFM = Read The Fucking Manual

4.2 Gestion du système

- `sudo commande` = exécuter une commande en tant que super-utilisateur
- `sudo apt update` = recharger la listes des programmes disponibles
- `sudo apt install nomduprogramme` = installer un programme
- `sudo apt remove nomduprogramme` = désinstaller un programme

4.3 Exploiter des fichiers texte

- `wget 'URL'` ou `curl 'URL'` = télécharge l'url
- `grep PATTERN FILE...` = recherche PATTERN dans FILE
- `cut -d' ' -f1,3 FILE` = sélectionne les champs 1 et 3 du fichier FILE, séparés par des espaces
- `sort FILE` = trie les lignes du fichier FILE
- `sort -u FILE` = trie les lignes du fichier FILE en supprimant les doublons
- `wc -l` = compte les lignes

4.4 Divers

- `echo 'texte'` = affiche le texte
- `date` = affiche la date
- `cal` = calendrier du mois en cours
- `cal mois annee` = calendrier du mois / annee
- `gzip fichier` = compresse le fichier vers fichier.gz
- `gunzip fichier.gz` = décompresse le fichier vers fichier
- (...) à compléter

5 Pipes et redirections

5.1 Redirections

La sortie d'une commande peut être envoyée dans un fichier

```
date > ladate  
cat ladate
```

On peut aussi **ajouter à la fin** du fichier au lieu de l'écraser :

```
echo 'je suis en NSI le' > nsi  
date >> nsi  
cat nsi
```

5.2 Pipes

On peut aussi envoyer la sortie d'une commande vers un autre commande :

```
wget 'https://raw.githubusercontent.com/datasciencedojo/datasets/master/titanic.csv'  
cat titanic.csv | grep female | wc -l
```