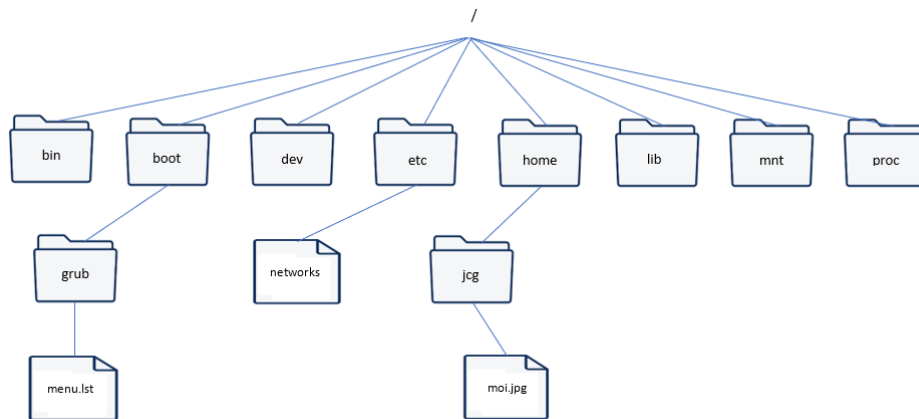


# TP – Connaître les commandes linux

Dans ce TP nous verrons les commandes de base à connaître permettant de gérer les répertoire et les fichier d'un système linux.

## Chemin relatifs – chemin absolu

Voici un exemple d'arborescence typique d'un système de fichier sous linux.



### A titre informatif :

/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

Dans un système d'exploitation, un chemin désigne l'emplacement d'un répertoire ou d'un dossier.

Il existe deux façons d'écrire le chemin menant à une ressource :

- Le chemin absolu :  
Il est décrit par l'ensemble des répertoires qu'il faut traverser depuis la racine ( / ). Par exemple, le chemin absolu du fichier moi.jpg est :  
/home/jcg/moi.jpg
- Le chemin relatif :  
Le chemin relatif est un chemin décrit depuis un autre répertoire. Par exemple, le chemin relatif du fichier moi.jpg depuis le répertoire home est :  
jcg/moi.jpg

Répertoire particulier :

- Le répertoire . désigne le répertoire lui-même. Par exemple, /home/. désigne home

- Le répertoire .. désigne le répertoire parent. Par exemple /home/.. désigne la racine
- ~ désigne le répertoire personnel de l'utilisateur connecté.
  1. Donner le chemin relatif de menu.lst depuis boot
  2. Donner le chemin absolu de networks
  3. Donner le chemin relatif de moi.jpg depuis mnt
  4. Donner le chemin relatif de networks depuis networks

## Les commandes de base – Gestion de fichier

### pwd – obtenir le chemin du répertoire courant

Lorsque l'on utilise un terminal, chaque commande est lancée depuis un répertoire. La commande pwd renvoie le chemin du répertoire dans lequel on se situe, celui à partir duquel on s'apprête à lancer une commande. Ce répertoire est appelé répertoire courant.

5. Utiliser la commande pwd pour connaître le répertoire dans lequel on se situe au lancement du terminal. Quel est ce répertoire ?

### mkdir – Créer un répertoire

La commande mkdir permet de créer un nouveau répertoire.

6. Taper la commande suivante pour créer un nouveau répertoire TNSI\_TP1 dans votre répertoire personnel.  
mkdir TNSI\_TP1

mkdir attend un argument, il s'agit du chemin (absolu ou relatif) du nouveau répertoire.

7. Utiliser la commande mkdir pour créer un répertoire enseignement\_scientifique dans le répertoire TNSI\_TP1.
8. Utiliser la commande mkdir pour créer un répertoire pc dans le répertoire enseignement\_scientifique
9. Utiliser la commande mkdir pour créer un répertoire svt dans le répertoire enseignement\_scientifique

### cd – changer de répertoire

La commande cd permet de changer de répertoire courant. C'est-à-dire de changer le répertoire depuis lequel on exécutera les futures commandes.

10. Taper la commande suivante pour entrer dans le répertoire TNSI\_TP1.  
cd TNSI\_TP1

Désormais, toutes les commandes que vous exécuterez le seront depuis le répertoire TNSI\_TP1.

La commande cd peut prendre 0 ou 1 arguments :

- Sans argument, la commande permet de placer dans le répertoire personnel de l'utilisateur
  - Avec un argument, on se place dans le répertoire passé en argument
11. Utiliser la commande cd pour vous placer dans votre répertoire personnel
  12. Créer un répertoire philosophie dans le répertoire TNSI\_TP1
  13. Se placer dans le répertoire que vous venez de créer.

### touch – Créer un fichier vide

La commande touch permet de créer un nouveau fichier vide.

14. Utiliser la commande suivante pour créer le fichier photosynthèse.txt dans le répertoire svt.  
touch ../enseignement\_scientifique/svt/photosynthèse.txt

(On utilise le chemin relatif depuis le répertoire philosophie jusqu'au nouveau fichier.)

15. Utiliser la commande touch pour créer un fichier platon.txt

**nano** : nano est un logiciel de traitement de texte en ligne de commande. Cela peut être très pratique lorsque l'on utilise un système d'exploitation ne disposant pas d'interface graphique.

16. Pour lancer nano et modifier le fichier platon.txt, suivez les étapes suivantes :
  - a. Lancer la commande suivante  
nano platon.txt
  - b. Ecrire la phrase « platon était un philosophe » dans le fichier
  - c. Ctrl+O pour sauvegarder
  - d. Ctrl+X pour quitter.
17. Ecrire la phrase « La photosynthèse a un rapport avec les plantes. » dans le fichier photosynthèse.txt

## ls – lister les éléments d'un répertoire

La commande ls permet de lister les éléments d'un répertoire.

18. Tester la commande suivante pour lister les éléments du répertoire courant  
ls
19. Se placer dans le répertoire TNSI\_TP1
20. Lister les éléments du répertoire courant

La commande ls peut prendre un argument en entrée, celui-ci correspond au chemin du répertoire que l'on veut explorer.

21. Tester la commande suivante permettant de lister les éléments du répertoire enseignement\_scientifique sans y être placé.  
ls /home/user/TNSI\_TP1/enseignement\_scientifique
22. Lister les éléments du répertoire svt.

La commande ls peut prendre certaines options. (Pour ajouter une option, on ajoute un '-' juste après le nom de la commande suivi du nom de la commande)

23. Tester la commande  
ls -l  
Que permet de faire l'option ?
24. Tester la commande  
ls -R  
Que permet de faire l'option ?

## cat – afficher le contenu d'un fichier

La commande cat permet d'afficher le contenu d'un fichier.

25. Tester la commande suivante permettant d'afficher le contenu du fichier platon.txt  
cat ./philosophie/platon.txt
26. Afficher le contenu du fichier photosynthèse.txt

## mv – Déplacer un élément

La commande mv permet de déplacer un élément. Pour cela, elle prend deux arguments. Le chemin du fichier à déplacer et le chemin du nouvel emplacement du fichier.

27. Se placer dans le répertoire enseignement\_scientifique
28. Créer un fichier atome.txt dans le répertoire svt
29. Utiliser la commande suivante pour déplacer le fichier atome.txt dans le répertoire pc

```
mv svt/atome.txt pc/atome.txt
```

30. Utiliser la commande mv pour simplement renommer le fichier atome.txt en atomes.txt  
(Pour renommer un fichier, il suffit de le déplacer dans le même répertoire que le répertoire initial et de modifier le nom du fichier)
31. Se placer dans le répertoire TNSI\_TP1 puis y créer un répertoire NSI.
32. Créer un répertoire spécialité
33. Déplacer le répertoire NSI dans le répertoire spécialité

## cp – Copier un élément

La commande cp permet de copier un fichier. Son fonctionnement est similaire à la commande mv. En effet, la commande attend deux arguments : le chemin du fichier source et le chemin du fichier destination.

34. Dans le répertoire NSI, créer le fichier notes.txt
35. Copier le fichier notes.txt dans les répertoires svt, pc et philosophie.

## rm – supprimer un élément

La commande rm permet de supprimer un élément.

36. Placez vous dans le répertoire svt
37. Utiliser la commande suivante pour supprimer le fichier photosynthèse.txt  

```
rm ./enseignement_scientifique/photosynthèse.txt
```
38. Supprimer le fichier atomes.txt

Pour supprimer un répertoire, il faut ajouter l'option -d.

39. Supprimer le répertoire pc.

Attention, il n'est pas possible de supprimer un répertoire non vide. Pour cela, il faut utiliser l'option -R permettant de supprimer le contenu du répertoire à supprimer de façon récursive.

40. En utilisant l'option -R, supprimer le répertoire spécialité.

## Permissions

Les droits d'accès définissent la possession d'un fichier ou d'un répertoire à un utilisateur et à un groupe d'utilisateurs. Ils gèrent aussi quelles actions les utilisateurs ont le droit d'effectuer sur les fichiers (lecture, écriture et exécution), selon qu'ils sont propriétaire du fichier, membre du groupe propriétaire du fichier ou ni l'un ni l'autre. La possession et la gestion des permissions associées s'effectuent individuellement avec chaque fichier.

Pour obtenir les informations concernant les permissions d'un fichier, il faut utiliser la commande ls -l.

```
[cbrassart@fedora philosophie]$ ls -l
total 4
-rw-r--r--. 1 cbrassart cbrassart 28 17 janv. 18:54 platon.txt
```

La première colonne d'information (-rw-r--r--) correspond aux permissions.

La seconde colonne d'information (1) correspond au nombre de liens.

La troisième colonne d'information (cbrassart) correspond au nom du propriétaire du fichier.

La quatrième colonne d'information (cbrassart) correspond au nom du groupe propriétaire du fichier.

La cinquième colonne d'information (28) correspond à la taille du document en octets.

La sixième colonne d'information (17 janv. 18 :54) correspond à la date de dernière modification.

La septième colonne d'information (platon.txt) correspond au nom du fichier.

## Comprendre les permissions

Comme on peut le constater, chaque fichier appartient à un utilisateur et à un groupe. C'est pourquoi les droits d'accès sont divisé ainsi :

- : Le premier caractère détermine le type de document. (d pour un répertoire par exemple)

rw- : les trois caractères suivants déterminent les droits du propriétaire sur son fichier.

r-- : les trois caractères suivants déterminent les droits des utilisateurs membres du groupe propriétaire sur le fichier.

r-- : Les trois derniers caractères détermine les droits de tous les autres utilisateurs sur le fichier.

Pour chacune de ces catégories, il existe donc trois droits :

r – correspond au droit de lecture. (Un utilisateur ne disposant pas du droit de lecture ne pourra pas lire le fichier (ou accéder au répertoire s'il s'agit d'un répertoire.)

w – Correspond au droit de d'écriture (Un utilisateur ne disposant pas du droit d'écriture ne pourra pas modifier le fichier (ou modifier les éléments au sein du répertoire s'il s'agit d'un répertoire.)

x – Correspond au droit d'exécution.

Pour modifier les droits d'un fichier on utilise la commande chmod.

Il existe deux possibilités :

- On définit les règles une à une, exemple  
chmod platon.txt u-w (retire le droit d'écriture (w) sur le fichier pour l'utilisateur propriétaire (u) )  
chmod platon.txt g+x (ajoute le droit d'exécuter (x) sur le fichier pour les membres du groupe propriétaire (g))  
chmod platon.txt o+r (ajoute le droit de lecture (r) sur le fichier pour tous les autres utilisateurs (o) )  
chmod platon.txt a+x (ajoute le droit d'exécuter (x) à tous les utilisateurs
- On définit les règles à l'aide du binaire.

Exemple :

rwX r-- --x      □      111 100 001

On définit la valeur décimale de chaque paquet de trois bits :

111 100 001      □      7 2 1

La commande 'chmod 721 platon.txt' permet de définir les droits du fichier sur rwX r-- --x

41. Supprimer vous le droit de modifier le fichier platon.txt
42. Lancer la commande nano platon.txt
43. Pourquoi cela ne fonctionne pas ?

Il existe une multitude d'autre commande linux existante.

- La commande help permet notamment de lister toutes les commandes disponibles.
- La commande man suivie d'une autre commande permet de d'obtenir la documentation de l'une des commandes.

# MEMO des commandes LINUX

## Fichiers et répertoires

### Généralités

Sur Linux, "*tout est fichier*" : les périphériques, les processus, les interfaces entre processus (socket)... Tout est organisé sous une seule arborescence de fichiers, commençant par le répertoire racine représenté par le symbole "/".

Les noms de fichiers et de répertoires sont sensibles à la casse et ne doivent pas contenir de caractères spéciaux (à l'exception du tiret bas et du point). Les répertoires sont vus comme des fichiers spéciaux qui contiennent une liste de fichiers et d'autres répertoires.

### Fichiers

#### Gestion

| Commande     | Description  | Syntaxe/Exemple   |
|--------------|--|---|
| <b>touch</b> | Crée un fichier vide ou actualise l'horodatage d'un fichier existant | <b>touch</b> <fichier><br><br><i># Crée un fichier datant d'il y a 2 jours</i><br><b>touch</b> -d "2 days ago" "monfichier.txt" |
| <b>cp</b>    | Copie un fichier vers un autre répertoire                            | <b>cp</b> <fichier_source> <fichier/répertoire_destination>   |
| <b>mv</b>    | Déplace/Renomme un fichier   | <b>mv</b> <fichier_source> <fichier_destination>  |
| <b>rm</b>    | Supprime un fichier  | <b>rm</b> <fichier>   |
|              | Désactive la confirmation de la suppression (f=force)                | <b>rm</b> -f <fichier>  |

### Affichage

| Commande | Description | Syntaxe/Exemple |
|----------|-------------|-----------------|
|----------|-------------|-----------------|

|             |  |  |
|-------------|--|--|
| <b>head</b> | <b>Affiche</b> les premières lignes d'un fichier   | <b>head</b> <fichier><br><br><i># Affiche les 5 premières lignes</i><br><b>head</b> -n 5 fichier.txt |
| <b>tail</b> | <b>Affiche</b> les dernières lignes d'un fichier   | <b>tail</b> <fichier><br><br><i># Affiche les 3 dernières lignes</i><br><b>tail</b> -n 3 fichier.txt |
|             | Affiche les nouvelles lignes <u>en continu</u> (f=follow)  | <b>tail</b> -f <fichier>   |
| <b>cat</b>  | <b>Affiche</b> le contenu d'un fichier   | <b>cat</b> <fichier>   |
| <b>nano</b> | <b>Édite</b> un fichier texte.<br>Cet éditeur simplifie grandement l'édition des fichiers. Ses commandes principales sont indiquées à l'écran, en voici d'autres : | <b>nano</b> <fichier>  |
| <b>vi</b>   | <b>Édite</b> un fichier texte.<br>Vi possède 2 modes : un mode édition et un mode commande. Voici quelques raccourcis pour l'utiliser :                            | <b>vi</b> <fichier>  |
| <b>vim</b>  | <b>Édite</b> un fichier texte.<br>Cet éditeur est basé sur Vi. Il utilise les mêmes raccourcis clavier et permet d'ouvrir des fichiers volumineux.                 | <b>vim</b> <fichier>   |

## Répertoires

| Commande   | Description                          | Syntaxe/Exemple           |
|------------|--------------------------------------|---------------------------|
| <b>pwd</b> | <b>Affiche</b> le répertoire courant | <b>pwd</b>                |
| <b>cd</b>  | <b>Modifie</b> le répertoire courant | <b>cd</b> <repertoire>    |
|            | <b>Crée</b> un répertoire            | <b>mkdir</b> <repertoire> |

|              |   |  |
|--------------|---|--|
| <b>mkdir</b> | Crée le répertoire <u>avec ses parents</u> si nécessaire  | <b>mkdir -p</b> <repertoire>                           |
| <b>rmdir</b> | <b>Supprime</b> un répertoire vide ou un fichier  | <b>rmdir</b> <fichier   repertoire>                    |
| <b>rm</b>    | <b>Supprime</b> un fichier ou un répertoire non vide  | <b>rm</b> <fichier   repertoire>                       |
|              | Supprime un fichier ou un répertoire non vide <u>en cascade et de façon forcée</u> (r=recursive, f=force) | <b>rm -r -f</b> <repertoire>                           |
| <b>mv</b>    | <b>Déplace/Renomme</b> un répertoire  | <b>mv</b> <repertoire_source> <repertoire_destination> |
| <b>ls</b>    | <b>Affiche</b> le contenu d'un répertoire   | <b>ls</b>  |
|              | <u>Présente les détails</u> : permissions, propriétaire, date de modification (l=long)                    | <b>ls -l</b>   |
|              | <u>Affiche</u> les 20 derniers fichiers créés   | <b>ls -ltr   top -n 20</b>                             |
| <b>cp</b>    | <b>Copie</b> un répertoire vers un autre répertoire.  | <b>cp</b> <repertoire_source> <repertoire_destination> |
| <b>tree</b>  | <b>Affiche</b> la structure d'un répertoire sous forme d'arbre  | <b>tree</b>  |

## Recherches chaîne

| Commande    | Description  | Syntaxe/Exemple                  |
|-------------|--|----------------------------------|
| <b>grep</b> | <b>Recherche</b> des caractères dans un fichier          | <b>grep</b> '<chaîne>' <fichier> |
|             | Afficher le <u>numéro de la ligne</u> trouvée (n=number) | <b>grep -n</b> '<chaîne>'        |

## Droits

### Généralités

Linux utilise un système de permissions basé sur des droits d'accès pour contrôler l'accès aux fichiers à travers un système de bits présent sur chaque fichier et répertoire (contrairement à Windows qui utilise un système de liste de contrôle d'accès, nommé ACL). Ces droits sont une manière de définir qui peut accéder à quel fichier et ce qu'il peut en faire. Il y a 3 catégories : le **propriétaire** du fichier, le **groupe propriétaire** et les **autres utilisateurs**.

Pour chaque catégorie, il y a trois types de permissions possibles : la **lecture** (r), l'**écriture** (w) et l'**exécution** (x). La lecture permet de voir le contenu d'un fichier, l'écriture permet de modifier le contenu et l'exécution permet de lancer le fichier s'il s'agit d'un programme exécutable.

Les droits sur un fichier sont représentés par une suite de 10 caractères :

- Les trois premiers caractères représentent les droits du propriétaire



- Les trois suivants ceux du groupe propriétaire
- Les trois derniers ceux des autres utilisateurs. Par exemple, si un fichier a les droits `rw-r--r--`, cela signifie que le propriétaire peut lire et écrire le fichier, que les membres du groupe propriétaire peuvent seulement le lire, et que les autres utilisateurs ne peuvent que le lire également.

### Permissions de fichiers et de répertoires

| Commande     | Description   | Syntaxe/Exemple                         |
|--------------|---|---|
| <b>chown</b> | <b>Modifie</b> le propriétaire d'un fichier ou d'un répertoire                    | <b>chown</b> <utilisateur> <fichier>    |
|              | Applique la modification à <u>tous les fichiers/répertoires</u> de l'arborescence | <b>chown</b> -r <utilisateur> <fichier> |
| <b>chmod</b> | <b>Modifie</b> les permissions d'un fichier/répertoire                            | <b>chmod</b> <valeur> <fichier>         |
|              | Ajoute (+) ou supprime (-) des permissions en lecture, écriture, exécution        | <b>chmod</b> +r +w -x <fichier>         |
| <b>chgrp</b> | Modifie le groupe d'appartenance d'un fichier/répertoire                          | <b>chgrp</b> <groupe> <fichier>         |

## Processus

### Lister les processus en cours d'exécution

| Commande   | Description  | Syntaxe/Exemple |
|------------|--|-----------------|
| <b>ps</b>  | <b>Affiche</b> les processus en cours                              | <b>ps</b>       |
|            | Utilise le format <u>détaillé</u>                                  | <b>ps</b> -ef   |
| <b>top</b> | <b>Affiche</b> les processus en cours en temps réel                | <b>top</b>      |
|            | Affiche <u>le nom complet de la commande</u> pour chaque processus | <b>top</b> -c   |
|            | Affiche les <u>threads</u> de chaque processus                     | <b>top</b> -H   |

### Gérer les processus

#### Fermer un processus

| Commande | Description | Syntaxe/Exemple |
|----------|-------------|-----------------|
|----------|-------------|-----------------|

|                |   |  |
|----------------|---|--|
| <b>kill</b>    | <b>Envoie</b> un signal à un processus pour demander son arrêt. | <b>kill</b> <pid>  |
|                | Utilisez l'option -9 pour forcer l'arrêt d'un processus.        | <b>kill</b> -9 <pid>   |
|                | Utilisez l'option -15 pour fermer proprement un processus       | <b>kill</b> -15 <pid/processus>  |
| <b>killall</b> | <b>Arrête</b> tous les processus selon leur nom                 | <b>killall</b> <processus><br><br><i># Fermeture de tous les process java</i><br><b>killall java</b> |

### Prioriser un processus

| Commande      | Description  | Syntaxe/Exemple  |
|---------------|--|--|
| <b>nice</b>   | <b>Définit</b> la priorité d'un processus à lancer. La priorité est paramétrée avec "-n".<br>-20 est la priorité la plus haute et 19 la plus basse | <b>nice</b> -n <priorite> <pid/processus><br><br><i># Démarre Firefox et lui affecte la priorité 10</i><br><b>nice</b> -n 10 firefox |
| <b>renice</b> | <b>Modifie</b> la priorité d'un processus existant selon son pid   | renice -n <priorite> <pid/processus><br><br><i># Affecte la priorité 5 au processus Firefox</i><br>renice -n 5 firefox               |