# I - Manipulation de fichiers

Pour chaque commande, recherchez à quoi elle sert, sa syntaxe, donnez au moins deux paramètres, et un exemple d’utilisation. Mettez à jour votre dictionnaire des commandes.

- man : permet de visionner le manuel d’une commande ou le manuel d’un fichier de configuration.

Par exemple : man [nom de la commande]  
 man tail

- touch : La commande [Linux](https://www.ionos.fr/digitalguide/serveur/know-how/linux-une-alternative-a-windows/) touch est utilisée pour **modifier les horodatages liés à la modification des fichiers ou répertoires individuels, ainsi qu’à l’accès à ces derniers**

Ex : touch [option] [fichier]

* **-a :** cette option vous permet uniquement de personnaliser le temps d’accès.
* **-m :** cette option vous permet uniquement de personnaliser l’heure de modification.

- cp : elle permet de copier un ou plusieurs fichiers vers un emplacement spécifié par l’utilisateur. À cet effet, deux informations sont tout particulièrement essentielles : **le nom de l’objet et l’objectif de l’opération**. La syntaxe de la commande CP est donc établie en conséquence :

:~$cp [chemin source] [chemin destination]  
:~$cp /home/morgane/documents /home/bob/documents

pour un dossier ajouter un –R :  
:~$cp -R/home/morgane/documents /home/bob/documents  
R pour reccursif  
  
pour un fichier :   
:~$cp /home/morgane/fichiers /home/bob/cahier = copie du fichier « fichiers » de morgane en fichier de nom « cahier » dans bob  
  
:~$cp /home/morgane/{fichier.txt,test.txt} /home/bob = copie les fichiers entre {} dans bob

- mv : déplacer un fichier ou un dossier  
même façon de faire que cp   
seule fois où il n’est pas nécessaire de préciser si c’est un fichier ou un dossier avec le -R

Pour renommer un fichier ou un dossier :   
:~$ mv /home/morgane/fichier.txt /home/morgane/test.txt = renomme le fichier fichier.txt en test.txt

- cat : Elle **affiche le contenu d'un fichier en sortie standard**.  
-A : montre tous le contenu

-b : numéroter les lignes non vides

- tac : concaténer et afficher des fichiers à l’envers  
-b : place le speparateur avant plutôt qu’apres  
-r : interpreter le separateur comme une expression rationnelle

- more : Filtre permettant de se déplacer dans un texte

-l : Ne pas faire de pause apres une ligne contenant un saut de page  
-f : compter les lignes logiques plutôt que celles de l’ecran

- less similaire a more mais avec plus d’options

- head : Affiche les 10 premières lignes de chaque Fichier sur la sortie standard.

-q : ne pas afficher les en têtes donnant le nom de fichiers  
-v : toujours afficher les en têtes donnant le nom des fichiers

- tail : Afficher la dernière partie de fichiers

-f : affiche les données ajoutées lorsque le fichier grossit

- nl : donne des numeros aux ligne par page

- diff : comparer des fichiers entre eux  
diff f1.txt f2.txt

- wc : affichez le nombre de lignes, de mots et de caractères d’un fichier. Cette commande peut également s’utiliser sur plusieurs documents, dont elle additionne alors les valeurs.

wc [Option] [Fichier]

- find :  Elle lance une **recherche récursive** dans laquelle un ou plusieurs répertoires sont explorés selon certains critères.

-name, -iname : Recherche par nom de fichier

-type : Recherche par type de fichier

- grep : **rechercher des chaînes spécifiques dans des fichiers ou pour filtrer ceux-ci** dans [Linux](https://www.ionos.fr/digitalguide/serveur/know-how/linux-une-alternative-a-windows/).

**grep [options] chaîne recherchée [fichier]**

**grep [options] [-e motif | -f fichier] [fichier]**

- split : La commande **split**permet d'éclater un fichier en fragments. Chaque fragment étant stocké dans des fichiers nommés

- rm : **La commande RM** est utilisée pour **supprimer des fichiers** et **des répertoires sous Linux** en ligne de commandes

-i : Invite et demande de confirmation avant chaque suppression

-f, –force : ignorer des fichiers et des arguments inexistants, jamais invite

# II - Manipulation de dossiers

Même consigne :

- mkdir : permet de créer un ou plusieurs dossiers dans un répertoire de travail en cours d’utilisation. Elle permet également de créer des hiérarchies de dossiers complexes.

$ mkdir [Option] NomDuRépertoire

- mkdir –p : Si nous utilisons l’option -p, la commande MKDIR est capable de créer les répertoires et n’affichera pas de message d’erreur.

- cp –R :   
Tous les répertoires (y compris les sous-répertoires) sont copiés de manière récurrente.

- mv : déplacer un fichier ou un dossier  
même façon de faire que cp   
seule fois où il n’est pas nécessaire de préciser si c’est un fichier ou un dossier avec le -R

Pour renommer un fichier ou un dossier :   
:~$ mv /home/morgane/fichier.txt /home/morgane/test.txt = renomme le fichier fichier.txt en test.txt

- rm –R :   
Supprimer les répertoires et leur contenu récursivement

- rmdir : supprimer définitivement un répertoire sous Linux

* chown (change owner) :

Vous pouvez changer ou transférer la propriété d’un fichier à un utilisateur donné avec « chown ». Tous les fichiers sous GNU/Linux sont en effet rattachés à un user spécifique.   
Cette commande prend en premier argument le nom du nouvel utilisateur et en second le nom du fichier dont on veut transférer la propriété

chown [-option] [nom proprio] [chemin dossier/fichier]

chown yann /home/dossier\_partage (Yann devient le propriétaire du dossier partage)

Pour changer la propriété de l’utilisateur ET du groupe :

chown [-option] [nom user proprio]:[nom grp proprio] [chemin dossier/fichier]

chown mathieu:coordo /home/planning (Mathieu devient le propriétaire du dossier planning et coordo devient le groupe propriétaire du dossier planning )

* chgrp :

Changer le groupe propriétaire d’un fichier

chgrp [-option] [nom groupe] [chemin dossier/fichier]

chgrp formateurs /home/dossier\_partage (le groupe « formateurs » devient le groupe propriétaire du dossier partage)

* + chmod

Les permissions de lecture (r), d’écriture (w) et d’exécution (x) de répertoires et de fichiers ont une grande importance sur Linux. Cette commande Linux permet de gérer et de changer ces permissions.

r = 4

w = 2

x = 1

chmod 770 /home/dossier\_partage/cours Permet de donner TOUS les droits de lecture (r), d’écriture (w) et d’exécution (x) du dossier cours à l’utilisateur (7) et au groupe (7) et aucun droit aux autres (0) (Other)

chmod 660 /home/dossier\_partage/cours/dev/php Permet de donner les droits de lecture (r), d’écriture (w) du fichier php à l’utilisateur (6) et au groupe (6) et aucun droit aux autres (0) (Other)

Si l’on veut cibler tous les dossiers de l’arborescence ajouter –R en option ex : chmod -R 770 /home/dossier\_partage/cours

u (user) g (groupe) o (other) a (all = ugo) r (read) w (write) x (exécute) + (ajouter) = (définir) – (supprimer)

chmod u+x /home/dossier\_partage/php (ajoute les droits d’exécution aux utilisateurs)

chmod o-rwx ou o= /home/planning (permet d’enlever tous les droits des autres utilisateurs (other)

chmod go+wx /home/morgane (ajoute aux groupe et aux autres la permission d’écrire (w) et d’exécuter (x)

chmod go+rw /home/morgane (ajoute aux groupe et aux autres la permission de lire (r) et d’écrire (w)

Haut du formulaire