

Prosit II Système d'exploitation : Commandes Linux de base

Classe : 1^{ère} année

Objectif :

Le présent document décrit brièvement quelques commandes linux ainsi que leurs options qui peuvent être utilisées dans le prosit 2.

Syntaxe générale d'une commande

Les commandes linux ont la syntaxe générale suivante :

- **\$** : c'est l'invite (prompt) de votre interpréteur de commande, et il ne faut pas le taper. Il indique que le système attend une commande.
- **Commande** : c'est le nom de la commande.
- **Options** : les commandes possèdent des options facultatives qui en modifient le comportement.
- **Paramètres** : les commandes prennent généralement des paramètres. Les paramètres sont les données d'entrées à la commande. Souvent, les paramètres sont des noms de fichier.

Commandes linux de base

Manipulation des répertoires (Change Directory)	
<i>Syntaxe</i>	<i>Description</i>
cd <i>chemin_repertoire</i>	Déplacement dans l'arborescence des répertoires vers le répertoire spécifié par le chemin donné à cd.
cd .	Positionnement sur le répertoire courant.
cd ..	Positionnement sur le répertoire parent du répertoire courant
cd ~	Positionnement sur le répertoire personnel de l'utilisateur connecté
cd /	Permet de remonter à la racine de l'ensemble du système de fichiers.

Lister les le contenu d'un répertoire (Liste Sorted)	
<i>Syntaxe</i>	<i>Description</i>
ls <i>chemin_repertoire</i>	Lister le contenu d'un répertoire.
ls -a <i>chemin_repertoire</i>	Lister le contenu du répertoire y compris les

	fichiers cachés.
ls -l <i>chemin_repertoire</i>	Lister le contenu d'un répertoire de manière détaillée
ls -d <i>chemin_repertoire</i>	Afficher les répertoires sans lister leurs contenus
ls -i <i>chemin_repertoire</i>	Afficher le numéro d'inode.

Copier fichiers / répertoires (copy)	
<i>Syntaxe</i>	<i>Description</i>
cp <i>chemin_fichier_source</i> <i>chemin_fichier_destination</i>	Copier le fichier source dans le chemin destination.
cp -i <i>chemin_fichier_source</i> <i>chemin_fichier_destination</i>	Avertir de l'existence d'un fichier du même nom et demande s'il peut ou non le remplacer.
cp -r <i>chemin_fichier_source</i> <i>chemin_fichier_destination</i>	Copier de manière récursive l'ensemble d'un répertoire et de ses sous répertoires.

Supprimer des fichiers / répertoires : (remove & remove directory)	
<i>Syntaxe</i>	<i>Description</i>
rm <i>chemin_fichier</i>	Supprimer un fichier.
rm -r <i>chemin_fichier</i>	Supprimer un répertoire et ses sous répertoires demande s'il peut ou non le remplacer.
rm -i <i>chemin_fichier</i>	Demander une confirmation avant la suppression du fichier.
rmdir <i>chemin_repertoire</i>	Supprimer un répertoire vide

Créer répertoires : (make directory)	
<i>Syntaxe</i>	<i>Description</i>
mkdir <i>chemin_repertoire</i>	Créer un répertoire.
mkdir -p <i>rep1/rep2/rep3</i>	Créer un répertoire et ses sous répertoires associés. (Créer toute une branche de l'arborescence)

Déplacer ou renommer un fichier: (move)	
<i>Syntaxe</i>	<i>Description</i>
mv <i>fichier1 fichier2</i>	Renommer <i>fichier1</i> en <i>fichier2</i> .
mv <i>fichier1 chemin</i>	Déplacer le fichier
mv -i <i>fichier1 fichier2</i>	Demande pour chaque fichier et chaque répertoire s'il peut ou non le déplacer.

Crée un lien (physique ou symbolique): (link)	
<i>Syntaxe</i>	<i>Description</i>
ln <i>fichier hardlink</i>	Créer un lien physique.
ln -s <i>fichier symboliclink</i>	Créer un lien "symbolique" (raccourci).

Recherche des fichiers (find, locate)																			
<i>Syntaxe</i>	<i>Description</i>																		
locate <i>fichier</i>	Rechercher ou localiser le fichier en donnant son chemin																		
find <i>répertoire option</i>	Rechercher des fichiers ou des répertoires dans «répertoire » en utilisant plusieurs options de recherche																		
Option de recherche avec « find »: <table border="1"> <tr> <td>-name</td><td>Recherche par nom de fichier.</td></tr> <tr> <td>-type</td><td>Recherche par type de fichier.</td></tr> <tr> <td>-user</td><td>Recherche par propriétaire.</td></tr> <tr> <td>-group</td><td>Recherche par appartenance à un groupe.</td></tr> <tr> <td>-size</td><td>Recherche par taille de fichier.</td></tr> <tr> <td>-atime</td><td>Recherche par date de dernier accès.</td></tr> <tr> <td>-mtime</td><td>Recherche par date de dernière modification.</td></tr> <tr> <td>-ctime</td><td>Recherche par date de création.</td></tr> <tr> <td>-perm</td><td>Recherche par autorisations d'accès.</td></tr> </table>		-name	Recherche par nom de fichier.	-type	Recherche par type de fichier.	-user	Recherche par propriétaire .	-group	Recherche par appartenance à un groupe .	-size	Recherche par taille de fichier.	-atime	Recherche par date de dernier accès .	-mtime	Recherche par date de dernière modification .	-ctime	Recherche par date de création .	-perm	Recherche par autorisations d'accès .
-name	Recherche par nom de fichier.																		
-type	Recherche par type de fichier.																		
-user	Recherche par propriétaire .																		
-group	Recherche par appartenance à un groupe .																		
-size	Recherche par taille de fichier.																		
-atime	Recherche par date de dernier accès .																		
-mtime	Recherche par date de dernière modification .																		
-ctime	Recherche par date de création .																		
-perm	Recherche par autorisations d'accès .																		

	-links	Recherche par nombre de liens au fichier.
--	---------------	--

GREP: Globally search for Regular Expression and print it	
<i>Syntaxe</i>	<i>Description</i>
grep <i>chaîne fichier</i>	Afficher toutes les lignes contenant « chaîne » du fichier.
grep -i <i>chaîne fichier</i>	Ignorer la différence entre majuscule et minuscule.
grep -v <i>chaîne fichier</i>	Afficher les lignes ne contenant pas la chaîne.
grep -n <i>chaîne fichier</i>	Chaque ligne contenant la chaîne est numérotée.

SORT	
<i>Syntaxe</i>	<i>Description</i>
sort <i>fichier</i>	Trier les lignes d'un fichier.
sort -n <i>fichier</i>	Trier par ordre numérique.

sort -r <i>fichier</i>	Trier inverse.
sort -R <i>fichier</i>	Trier aléatoirement.
sort -o <i>fichier_de_sortie</i>	Ecrire dans le fichier_de_sortie plutôt que sur la sortie standard.
sort -t [<i>séparateur</i>]	Utiliser le « séparateur » afin de distinguer les colonnes pour rechercher la clé de tri sur chaque ligne.
sort -t [<i>séparateur</i>] -k [<i>n</i>]	Trier la «n»ème colonne selon le séparateur
uniq <i>fichier</i>	Supprimer les doublons (applicable que sur les fichiers triés)

WC: Word Count	
<i>Syntaxe</i>	<i>Description</i>
wc <i>fichier</i>	Affiche le nombre de lignes, nombre de mots et le nombre d'octets du fichier.
wc -l <i>fichier</i>	Affiche le nombre de lignes d'un fichier.
wc -w <i>fichier</i>	Affiche le nombre de mots d'un fichier.

wc -c <i>fichier</i>	Affiche le nombre d'octets dans un fichier.
wc -m <i>fichier</i>	Affiche le nombre de caractères d'un fichier.

CAT : conCATenate	
<i>Syntaxe</i>	<i>Description</i>
cat <i>fichier</i>	Afficher le contenu d'un fichier
cat <i>fichier1, fichier2</i>	Afficher le contenu des fichiers fichier1 et fichier2 concaténés.
cat -n <i>fichier</i>	Afficher le contenu du fichier avec des numéros de ligne.
nl <i>fichier</i>	Afficher le contenu du fichier avec des numéros de ligne. (sans tenir en compte les lignes vides)

HEAD / TAIL	
<i>Syntaxe</i>	<i>Description</i>
head <i>fichier</i>	Affiche par défaut les dix premières lignes d'un fichier
head -n <i>fichier</i>	Affiche les n premières lignes du fichier.
tail <i>fichier</i>	Affiche par défaut les dix dernières lignes d'un fichier
tail -n <i>fichier</i>	Affiche les n dernières lignes du fichier.

CUT	
<i>Syntaxe</i>	<i>Description</i>
cut <i>fichier</i>	Permet de filtrer (supprimer) une partie de chacune des lignes d'un fichier.
cut -d [délimiteur] <i>fichier</i>	L'option -d permet d'exprimer le caractère séparateur de champ [délimiteur]. Le caractère séparateur par défaut est la tabulation.
cut -f <i>fichier</i>	Permet à définir <nb> champ à garder

cut -c[n] fichier	Permet d'afficher uniquement le «n»ème caractère de chaque ligne d'un fichier.
cut -d[délimiteur] -f[n] fichier2	Afficher le «n»ème champs selon le délimiteur [délimiteur] de chaque ligne du fichier.

TR : TRanslate	
<i>Syntaxe</i>	<i>Description</i>
tr "[A-Z]" "[a-z]"	Permet de convertir une chaîne de caractères.
tr -d caractère	Effacer le caractère spécifié.

Gestion des utilisateurs	
<i>Syntaxe</i>	<i>Description</i>
sudo useradd nom_utilisateur	Créer un nouvel utilisateur.
sudo groupadd nom_groupe	Ajouter un groupe.

sudo userdel <i>nom_utilisateur</i>	Suppression d'un utilisateur.
sudo groupdel <i>nom_groupe</i>	Suppression d'un groupe.

Gestion des droits	
<i>Syntaxe</i>	<i>Description</i>
chmod <i>options modes fichier</i> chmod [<i>u g o a</i>] [<i>+ - =</i>] [<i>r w x</i>] <i>nom_du_fichier</i> chmod -R [<i>u g o a</i>] [<i>+ - =</i>] [<i>r w x</i>] <i>nom_du_répertoire</i>	Changer les permissions d'accès d'un fichier ou d'un répertoire.
chmod [<i>valeur *</i>] [<i>nom du fichier</i>]	Changer les permissions d'accès d'un fichier ou d'un répertoire.
chown [<i>propriétaire</i>] [<i>nom du fichier</i>]	Permet de modifier le propriétaire d'un fichier.
chgrp [<i>groupe propriétaire</i>] [<i>nom du fichier</i>]	Permet de modifier le groupe propriétaire d'un fichier.
umask [<i>valeur umask**</i>] [<i>nom du fichier</i>]	définir les droits par défaut d'un fichier ou un répertoire à sa création

Plus d'information : <https://www.leshirondellesdunet.com/chmod-et-chown>

* Vous avez une autre méthode pour vous servir de la commande **chmod**. On considère que **r=4**, **w=2** et **x=1**, si vous avez un fichier avec les droits suivants **-rw-rw-rw-**, pour les droits utilisateurs vous avez **(r=)4+(w=)2=6**, de même pour le groupe et les autres. Donc **-rw-rw-rw-** est équivalent à **666**. En suivant la même règle **rw-rw-rw-** est équivalent à 754. Pour mettre un fichier avec les droits **-r--r--r--** vous pouvez taper : `chmod 444 nom-fichier`

** `umask 002`

- Pour les fichiers :

Avec cet exemple de mask positionné à 002, les fichiers créés auront par défaut les droits `rw-rw-r--`, cela s'explique avec l'algèbre booléen:

Quand on crée un fichier, les droits de celui-ci sont 666 (soit `rw-rw-rw`) **ET** le complément du masque (NON masque)

Si on note ça en bit, cela fait :

110 110 110 pour les `rw-rw-rw-`

111 111 101 le complément du masque de 002 : masque (000 000 010)

110 110 100 pour 666 et Non masque (résultat ET logique)

`rw- rw- r--`

- Pour les répertoires :

Avec cet exemple de mask positionné à 002, les répertoires créés auront par défaut les droits `rw-rw-rw-x`, cela s'explique avec l'algèbre booléen:

Quand on crée un répertoire, les droits de celui-ci sont 777 (soit rwxrwxrwx) **ET** le complément du masque (NON masque)

Si on note ça en bit, cela fait :

111 111 111 pour les rwxrwxrwx

111 111 101 le complément du masque de 002 : masque (000 000 010)

111 111 101 pour 777 et Non masque (résultat ET logique)

rwx rwx r-x

Autres commandes	
<i>Syntaxe</i>	<i>Description</i>
pwd	Où suis-je ?
man <i>nom commande</i>	Rechercher des informations sur une commande
touch <i>nouveau fichier</i>	Création d'un nouveau fichier
echo <i>ligne</i>	Permet d'afficher une ligne dans le terminal.
vi <i>fichier</i>	Lancer un éditeur de texte en mode texte plein écran pour éditer le fichier
nano <i>fichier</i>	Lancer un éditeur de texte en mode texte plein écran pour éditer le fichier