

# Se familiariser avec les commandes linux

**Objectif :** Le but de cette partie est la prise en main des commandes de base de l'environnement Linux.

## Enoncé :

1. Qu'est-ce que le shell ?

... interpréteur de commande / interface entre S.E et user

2. Identifier les répertoires suivants :

... / (racine) /usr (logiciels)

... /etc (paramètres)

... /home (dossier utilisateur)

3. Compléter le tableau

Commande	A quoi sert	exemple
cd	Change le répertoire courant	cd .. cd /usr
ls	affiche le contenu d'un rep	ls [ affiche contenu d'un rep ] ls .. copie le fichier
cp	Copie un ou plusieurs fichiers.	cp fichier1 fichier2 cp .. /f1 .. /rep1 / f2
rm	Efface un ou plusieurs fichiers	rm .. /rep1 / f1 (un fichier) / f2 rm -r .. /rep1 (supprime tout le rep1)
mkdir	Créer un répertoire (vide)	mkdir rep1 rep2 rep3
rmdir	Supprimer un rep vide	rmdir rep1
mv	Déplace ou renomme un fichier	mv f1 f2 mv .. /f1 .. /rep1 / f2
cat	affiche le contenu d'un fichier	cat f1 cat f1 f2
man	affiche help d'une cmd	man ls man .. ls
touch	crée un fichier vide	touch .. /f1 touch .. /f2 .. /rep1 / f3

4. Afficher le listing d'un repertoire en couleurs

... ls -la

pwd : print working directory  
 ( dir rep cour )  
 ~ : rep home  
 ls : liste des fichiers qui existent dans mon repertoire  
 ls -l : affiche les fichiers cachés  
 ls -la : affiche les fichiers cachés  
 cp : copie un ou plusieurs fichiers  
 cp .. / f1 : / rep 1 / f1  
 Source destination  
 mv : déplace ou renomme un fichier  
 mv f1 f2 : mv . / f1 . / rep 1 / f2  
 cat : affiche le contenu d'un fichier  
 cat f1 : cat f1 f2  
 man : affiche help d'une cmd  
 touch : crée un fichier vide  
 touch . / f1 : touch . / f2 . / rep 1 / f3  
 rm : supprime un ou plusieurs fichiers  
 rm f1 : rm -r . / rep 1 (supprime tout les fichiers)  
 rmdir : supprime un repertoire vide  
 rmdir rep1  
 mkdir : crée un repertoire (vide)  
 mkdir rep1  
 cd : change le repertoire courant  
 cd .. : cd un

2. Identifier les répertoires suivants :

rep (dossier) courant ( la )  
 rep parent  
 rep home

3. Compléter le tableau

Commande	A quoi sert	exemple
cd	Change le repertoire courant	cd .. cd un
ls	affiche le contenu d'un rep	ls ( affiche contenu d'un rep ) ls .. copie le contenu courant
cp	Copie un ou plusieurs fichiers.	cp f1 f2 cp .. / f1 . / rep 1 / f2
rm	Efface un ou plusieurs fichiers	rm . / rep 1 / f1 (un fichier) / f2 rm -r . / rep 1 (supprime tout les fichiers)
mkdir	Créer un repertoire (vide)	mkdir rep1 mkdir rep1 rep2 rep3
rmdir	Supprimer un rep vide	rmdir rep1
mv	Déplace ou renomme un fichier	mv f1 f2 mv . / f1 . / rep 1 / f2
cat	affiche le contenu d'un fichier	cat f1 cat f1 f2
man	affiche help d'une cmd	man ls man f1
touch	Crée un fichier vide	touch . / f1 touch . / f2 . / rep 1 / f3

4. Afficher le listing d'un repertoire en couleurs

ls -la



fichier / rep  
 cat / affiche les données d'un fichier  
 man / affiche les données d'un fichier  
 5. Afficher aussi les fichiers cachés (commençant par un point) :  
 - fichiers cachés : commencent par un point  
 - fichiers normaux : ne commencent pas par un point  
 - type de l'élément :  
 - fichiers ordinaires = fichiers de données  
 d : directory (rep / dossier)  
 l : liens  
 c : périphérique caractéristique  
 b : périphérique bloc  
 touch B  
 chmod 400 B  
 rwx - r - -  
 u - g - o -  
 r : read  
 w : write  
 x : execute  
 user : groupe : other

6. Afficher le listing détaillé

ls -l

xrw - r - -  
 110 100 100  
 6 4 4

Pour afficher les fichiers d'un répertoire en triant sur la date de mise à jour des fichiers

7. Afficher les fichiers les plus récents en premier

ls -lt

8. Afficher les fichiers les plus vieux en premier

ls -ltr

Dans les systèmes d'exploitation dérivés d'Unix, le codage des droits se fait sur 9 bits groupés par 3 bits. Ces droits sont codés en un entier. Pour ce faire, on convient de la correspondance : r = 4; w = 2 et x = 1.

9. A quels droits correspondent les entiers 751; 521; 214 et 150 ?

rwx | groupe | other  
 user

751 : 521 : 214 : 150  
 rwx r-x - - x : o-x -w- : -w- -x r - - : -x | 1  
 r-x - - -

d : dossier  
 - : fichier ordinaire  
 l : lien

c : b spécial  
 b : b spécial

10. Par quels entiers sont codés les droits rw-r-r- et rwxr-xr-x ?

chmod : cmd qui permet  
 de changer  
 des droits

(110) (100) (100) (111) (101) (101)  
 6 4 4 7 5 5

chmod  
 644 : fichier/dossier  
 option

La commande ls -l toto.sh affiche :

-r-x-r--r-- 1 john staff 128 18 mai 11:56 toto.sh

11. À qui appartient ce fichier ? user

toto.sh appartient à john (user)

12. Qui peut écrire dans le fichier ?

personne (w manquant pr user, groupe et other)

touch f, // creation d'un fichier f, vide

$P_s$   $P$   
 $-$   $nw - n - 1$  user, user 0 oct 2023  
 $664$   $chmod 0+x f,$   
 $f,$  ajout de x pr user  $ugo =$

$chmod 764 f,$  argument  
 $mbactop$  conditionnée : je connais les anciens droits

$u$   $g$   $o$   $qui =$  affectation  $choix$   
 $ajout$   $optim$   $mais$   $n$   
 $w$   
 $x$

des données jouées

$fichiers$   $echo \text{ " " } > f,$

$echo \text{ " " } >> f,$

éditeur  $\left\{ \begin{array}{l} \text{textuel : vi } f, \\ \text{graphique gedit } f, \end{array} \right.$

2/ élimer droit write du groupe

$chmod g-w f,$

3/ Donner que les dls net x pr other

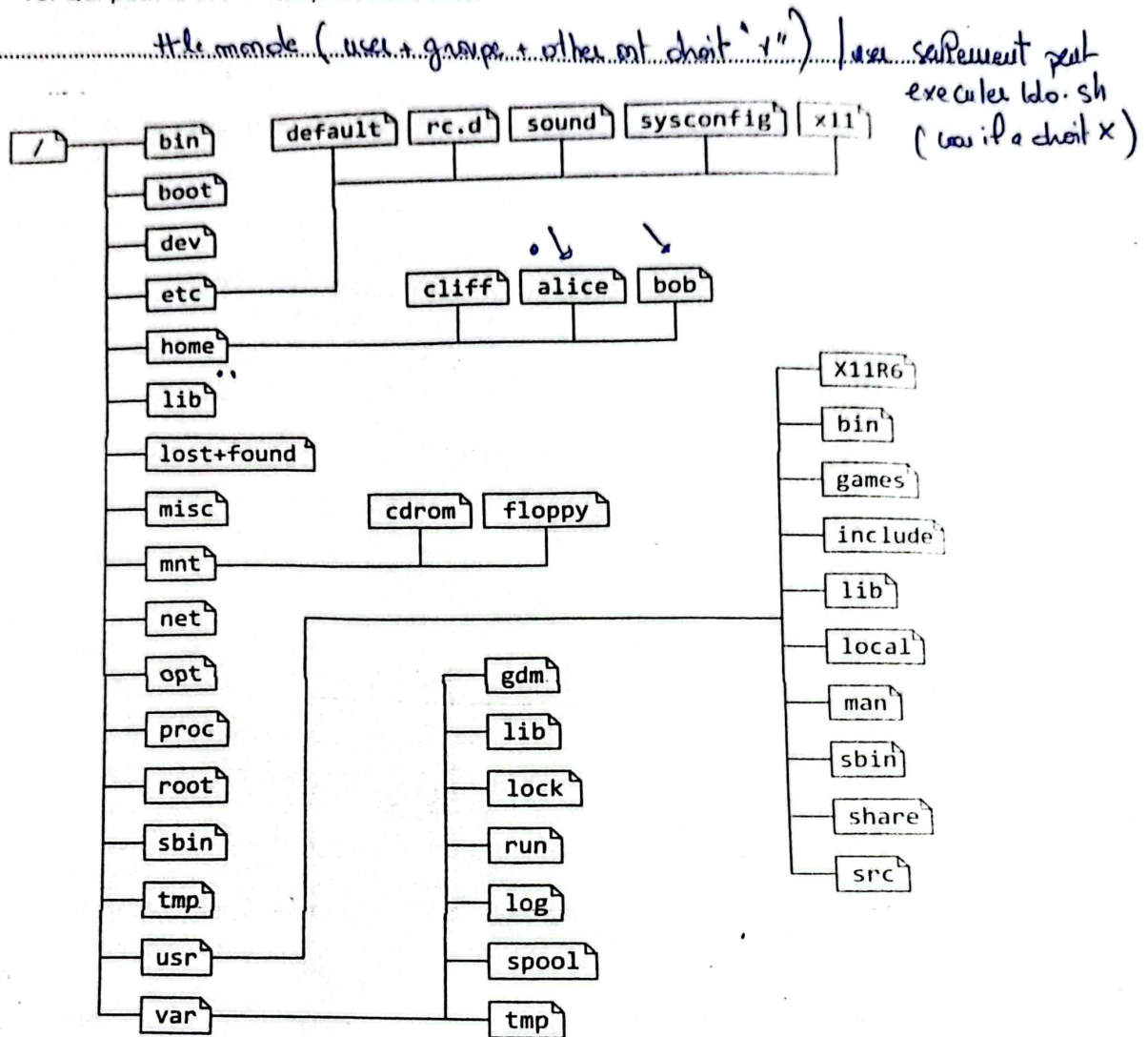
$chmod o = nx f,$

$ugo = \cancel{nw} \times \cancel{nw} = n$

$u = nw \times u \quad g = nw \quad o = n$

$ugo = nw - \underbrace{nw - nw - nw}_{\text{and}} - \underbrace{f,}_{\text{arg}} = 0$

13. Qui peut le lire ? Qui peut l'exécuter ?



14. Proposer une commande qui permette de se déplacer du répertoire HOME de Alice à celui de Bob :

a. en utilisant un chemin relatif ;

*Handwritten answer:* cd ../.. / bob

b. en utilisant un chemin absolu.

*Handwritten answer:* cd /home / bob

15. Alice est à la racine /. Proposer trois commandes qui peuvent lui permettre de se déplacer dans son répertoire d'accueil (HOME).

*Handwritten answer:* cd Alice / home

*Handwritten answer:* cd

16. Bob est dans son HOME. Aidez-le à :

a. lister le contenu de son HOME ;

ls

b. lister le contenu de son HOME y compris les fichiers et répertoires cachés.

ls -a

c. lister le contenu du répertoire share sans quitter son HOME.

ls -o / / user / share / (relatif)

ls / user / share (absolu)

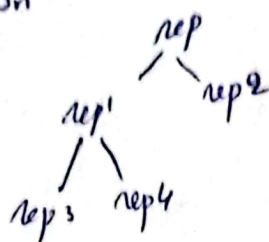
**Bon travail**



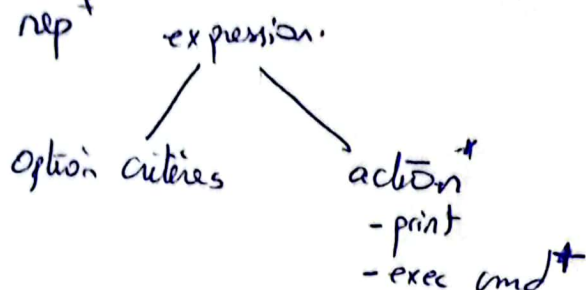
cmd opt<sup>o</sup> argument

find repetitive expression

find rep

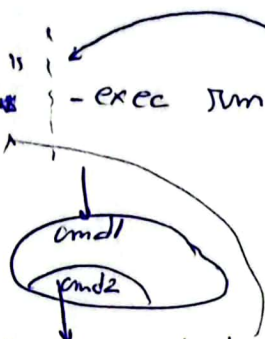


find rep<sup>+</sup>



find rep - name "f" { -exec run ? } ; |

cmd1 | cmd2



\* : 0 ou plusieurs instances (y compris 0)  
+ : 1 ou plusieurs instances

? : 1

[ ] : interval  
[a-z], [a-du-z], [a-c b p] (w maj + ou bin b minuscule)  
[a-z A-Z]

{ }

[a c b l] ≡ { a, c b, l }

\* commence par  
[a-z A-Z]\*

1 [a-z A-Z] [0-9]<sup>+</sup>

maj & min -> chiffres

## TD les filtres

### Commande « find »

1. Cherchez un fichier dans le répertoire racine qui contient le terme "net".

`find / -type f -name "*net*"`

2. Chercher tous les fichiers dont les noms commencent par une majuscule ou une

`find / -type f -name "[A-Z]*" -o -name "[a-z]*" -o -name "[0-9]*"`

3. Chercher les fichiers qui ont été modifiés il y a plus de 30 jours ? Il y a 30 jours ? Il y a moins de 30 jours ?

`find / -type f -mtime +30` *il y a plus de 30 jrs*

`find / -type f -mtime -30`

`find / -type f -mtime -30` *il y a au plus 30 jrs*

4. Chercher le fichier a été modifié plus récemment qu'un autre fichier donné ?

`find / -type f -newer "fichier.txt"`

5. Cherchons les fichiers modifiés plus récemment que bofichier.txt, mais il y a plus de 5 jours

`find / -type f -newer "bofichier.txt" -mtime +5`

6. On recherche un fichier modifié il y a moins de 12 jours et dont la taille soit supérieure à 30 K

`find / -type f -mtime -12 -size +30k`

7. Chercher dans /usr les fichiers dont la taille dépasse 1Mo et dont les droits sont fixés à 755

`find /usr -type f -size +1m -perm 755`

8. Cherchez dans toute l'arborescence les fichiers dont le nom se termine par .c, redirigez les erreurs vers le fichier poubelle /dev/null

`find / -type f -name "*.c" 2>/dev/null`

9. Trouvez tous les fichiers modifiés entre 10 et 50 jours auparavant.

`find / -type f -mtime +10`

10. Trouver des fichiers qui ont une taille > 10 Mo et < 40 Mo.

`find / -type f -size +10240k -size -40240k` *10 < date de modif < 50*

Bon travail



éditeur de txt

textuelle

graphique

gedit nom\_fichier

éditeur

vi , nom

md interne à utiliser

exp vi nom\_fichier

just text " "

loinsert

vi resume

echap

:wq ) sauvegarder et quitter

echo " ligne 1

ligne 2

dernière ligne " >> resume



grep "aH" resume

grep "test" resume

"test"

"test"

"test"

nom\_fichier

echo "ligne 1"

ligne 2

dernière ligne" >> resume

grep -c "test" resume : nbr de lignes contenant "test"

=> nbr d'occurrence il faut faire - script

vi nom\_fichier

vi resume

↳ insert

echap

:wq) sauvegarder et quitter