

Résumé des commandes UNIX

Ls		
	Ls NomDuRepertoire	Afficher le contenu du répertoire
-A	Ls -A NomDuRepertoire	Afficher le contenu du répertoire sauf . et ..
-R	Ls -R NomDuRepertoire	Lister récursivement : le répertoire et ses sous répertoires
-t	Ls -t NomDuRepertoire	Afficher selon la date de dernière modification (newest first)
-r	Ls -r NomDuRepertoire	Inverser l'ordre d'affichage
-c	Ls -c NomDuRepertoire	Afficher selon dernière modification de statut
-i	Ls -i NomDuRepertoire	Affichage trié selon inode
-l	Ls -l NomDuRepertoire	Affichage détaillé des fichiers (statut) : <ol style="list-style-type: none"> 1. Le type et les droits d'accès 2. Le nombre de liens physiques 3. Le nom du propriétaire du fichier 4. Le nom du groupe du fichier 5. Taille en octets 6. 7 et 8 : mois, jour et heure de la dernière modification 9. le nom du fichier
-F	Ls -F NomDuRepertoire	Permet d'identifier plus vite le type de fichiers : / pour les répertoires @ pour les liens symboliques

Mkdir		
	Mkdir rep1 rep2 rep3	Créer plusieurs répertoires en une seule ligne de commande
	rmdir rep1 rep2 rep3 rm -r rep1 rep2 rep3	Supprimer plusieurs répertoires en une seule ligne de commande
-p	Mkdir -p rep1/rep2	Créer à la fois un répertoire et un sous-répertoire
Touch	Touch f.txt	Créer un fichier texte

Cp		
-r	Cp -r NomDuRepertoire	Faire une copie du répertoire et de l'ensemble de ses sous-répertoires
	Cp -r rep1 rep2	<ul style="list-style-type: none"> • Si rep2 existe : rep1 sera copié dans rep2 et gardera son nom « rep1 » • Sinon, rep1 sera copié dans le répertoire courant et la copie sera nommée « rep2 »
	Cp fic1 rep	<ul style="list-style-type: none"> • Si rep contient un fichier nommé « fic1 » ce fichier sera remplacé par fic1 • On peut ajouter -i pour que le système demande une confirmation avant d'écraser « fic1 »
	Cp f1 f2 f3 rep	Copier plusieurs fichiers dans un répertoire
	Cp -r f1 f2 f3 rep1 rep	Copier aussi rep1 dans rep

Mv

	Mv fichier1 fichier2	<ul style="list-style-type: none"> • Si fichier2 n'existe pas : fichier1 sera renommé fichier2 • Sinon fichier2 sera écrasé et fichier1 renommé
	Mv rep1 rep2	<ul style="list-style-type: none"> • Si rep2 existe : rep1 sera déplacé à l'intérieur de rep2 sans être renommé • Sinon rep1 sera déplacé et renommé
-i	Mv -i fic1 fic2	Mode interactif : demande à l'utilisateur une confirmation avant d'écraser fic2, on peut répondre par y (yes) ou une touche qqc pour (No)
-f	Mv -f fic1 fic2	<ul style="list-style-type: none"> • Mode forcé : écrase le fichier sans rien demander • Si on utilise les deux options -i et -f dans une seule commande, c'est celle qui est écrite en dernier qui sera prise en considération

Chmod		
	Chmod 754 f.txt Chmod u+x, g+x f.txt Chmod ug+x f.txt	Ajouter l'exécution au propriétaire et au groupe : le fichier est créé avec les droits 644 puisque l'umask par défaut est 022
	Chmod o=g f.txt	Affecter les droits du groupe aux autres
	Chmod o=- f.txt	Aucun droit pour les autres
-R	Chmod -R 732 rep	Pour que le changement soit appliqué au rep et son contenu

Les Jokers		
	rm essai[1-4]	Supprime essai1 essai2 essai3 et essai4
	mv *.c C	Regrouper tous les fichiers avec l'extension .c dans un répertoire C
	Mv *[aeiouAEIOU][aeiouAEIOU] [aeiouAEIOU]* Rangement	Regrouper dans le répertoire « Rangement » les fichiers dont le nom contient 3 voyelles à la suite
	Mv *[aeiouAEIOU]*[aeiouAEIOU]*[aeiouAEIOU]* Rangement	Regrouper dans le répertoire « Rangement » les fichiers dont le nom contient 3 voyelles
	Ls /bin/ a[st]* Ls /bin grep '^a[st]'	Afficher la liste des fichiers dans /bin dont le nom commence par a et la 2ème lettre est "s" ou "t"
	Ls /bin/ *un*t Ls /bin grep 'un.*t\$'	Fichiers dont le nom contient « un » et se termine par « t »
	Ls /bin/*m*m* Ls /bin grep '.*m.*m'	Fichiers dont le nom contient au moins 2 "m"
	Ls /bin/ ????* Ls grep '.....*' Ls grep '\.{4,}\'	Fichiers dont le nom contient au moins 4 caractères

Ls /bin/ ????	Fichiers dont le nom contient exactement 4 caractères
Ls grep '^....\$'	
Ls grep '^.{4,}\$'	
Ls /bin/ [0-9]*[0-9]	Fichiers dont le nom commence et finit par un chiffre
Ls /bin grep '^[0-9].[0-9]\$'	

Find		
-name	Find -name "[aA]*[3-6] "	Chercher les fichiers dont le nom commence par 'a' ou 'A' et se termine par un chiffre entre 3 et 6
-mtime	Find -mtime +7 Find -mtime 7 Find -mtime -7	Chercher des fichiers modifiés il y a : <ul style="list-style-type: none"> • Plus de 7 jours • 7 jours • Moins de 7 jours
-newer	Find -newer file	Chercher des fichiers modifiés plus récemment que file
-type d	Find -type d	Les fichiers recherchés sont des répertoires
-size	Find -size +20c Find -size +20 Find -size +20w	Chercher les fichiers dont la taille est : <ul style="list-style-type: none"> • > 20 octets • > 20 blocs (bloc=512 oct) • > 20 words (word=2 oct)
	Find / -mtime +30 -name "*~" -exec rm {} \	<ul style="list-style-type: none"> ▪ Supprimer tous les fichiers dont le nom se termine par '~' ▪ 'xargs' sert à passer les fichiers trouvés à rm en argument
	Find / -mtime +30 -name "*~" -delete	
	Find / -mtime +30 -name "*~" xargs rm	
-perm mode	Find -perm 755 -type f Find -type f -perm u=rwx, g=rx, o=rx	Chercher à partir du répertoire courant les fichiers réguliers dont les droits d'accès sont exactement rwxr-xr-x
-perm /mode	Find -perm / u=x, g=x, o=x Find -perm /111	Chercher les fichiers réguliers executables: qui ont 'x' pour u ou g ou o
-perm -mode	Find -perm -u=x, g=x, o=x Find -perm -111	Chercher les fichiers réguliers qui ont au moins tous les bits spécifiés à 1 : <ul style="list-style-type: none"> ▪ Si un fichier a un bit de moins, il ne sera pas affiché ▪ Si un fichier a un bit de plus, il sera affiché

Cat		
	Cat f1 f2	Afficher le contenu des deux fichiers l'un après l'autre
	Cat>file	Pour écrire dans file, une fois la saisie terminée on tape ctrl+D
	Cat fic1 fic2 >> fic3	Ajouter le contenu de fic1 et fic2 à la fin de fic3 en gardant son contenu
	Cat fic1 fic2 > fic3	<ul style="list-style-type: none"> ▪ Mettre le contenu de fic1 et fic2 à la fin de fic3 en écrasant son contenu ▪ Si fic3 n'existe pas, cette commande et la précédente seront équivalentes et elles permettent de créer fic3 et mettre le contenu

		de fic1 et fic2 dedans. ▪ Si fic1 ou fic2 n'existe pas, on copie celui qui existe et on génère un message d'erreur
-n	Cat -n fic1 > fic2	Créer un fichier fic2 contenant les lignes de fic1 numérotées

Head – Tail	
Head file	Affiche par défaut les 10 premières lignes
Head -15 file Head -n +15 file	Afficher les 15 premières lignes de file
Head -n -k file	Affiche les L-k premières lignes de file, avec L le nbr total des lignes
Tail -15 file Tail -n -15 file	Afficher les 15 dernières lignes de file
Tail -n +15 file	Afficher toutes les lignes à partir de la 15 ^{ème}
Head -20 file tail -n +15 Tail -n +15 file head -5	Afficher les lignes 15 à 20
Head -n -3 file tail -n +15 Head -9 file tail -n +5	Afficher les lignes 5 à 9 d'un fichier de 12 lignes
Head -5 file tail -n +5	Afficher la 5 ^{ème} ligne de file

Grep		
-A x		Permet d'obtenir x lignes après le mot recherché
-B x		Permet d'obtenir x lignes avant le mot recherché
-C x		Permet d'obtenir x lignes avant et après le mot recherché
-c -occ	Grep -c 'motif' file Grep -c 'motif' file Grep 'motif' file wc -l	Nombre d'occurrences de motif dans file
-i		Permet d'ignorer la casse
-l		Afficher les fichiers où le mot existe
-v		Afficher les lignes où le mot cherché ne figure pas
-L		Afficher les noms de fichiers ne contenant pas le mot recherché
-w		Chercher le motif et non pas ses dérivées : Si on cherche 'travail', ne pas afficher les lignes contenant 'travailler'
-E	Grep -E -n 'occ1 occ2' file Grep -n 'occ1 \ occ2' file	Chercher plusieurs mots à la fois
	Grep '^ [Aa]' file Grep -i '^a' file	Chercher toutes les lignes qui commencent par « a » ou « A »
	Grep 'rs\$' file	Chercher toutes les lignes finissant par « rs »

	Grep '[0-9]' file	Chercher toutes les lignes contenant au moins un chiffre
	Grep '^ [A-Z]' file	Chercher toutes les lignes commençant par une majuscule
	Grep '!\$' file	Chercher les lignes finissant par « ! »
	Ls /bin grep '^a[st]'	Liste des fichiers dans /bin dont le nom commence par a et la 2 ^{ème} lettre est s ou t
	Ls /bin grep 'gre \ st' Ls /bin grep -E 'gre st' Ls /bin grep -e gre -e st	Fichiers dont le nom contient « gre » ou « st »
	Ls /bin grep '^ (^m)*m (^m)*m (^m)*\$'	Fichiers dont le nom contient exactement 2 lettres "m"
	Ls /bin grep 'm.*m'	Fichiers dont le nom contient au moins 2 lettres "m"
	Ls /bin grep '^ (^0-9)[^0-9][^0-9][^0-9][^0-9]*\$' Ls /bin grep '^ (^0-9){4,}\$'	Fichiers dont le nom contient au moins 4 caractères et aucun chiffre
	Grep -v '^\$' file Grep '.' File	Eliminer les lignes vides (n'élimine pas les lignes ne contenant que des espaces)
	Grep -v '^[[:space :]] [[:space :]]*\$' file	Eliminer les lignes ne contenant que des blancs
	Grep -v '^[[:space :]]*\$' file	Eliminer toutes les lignes blanches
	Grep -v '^[[:space :]]*\$' file1 > file2	Écrire les lignes non vides (pour l'oeil) de file1 dans file2
	Sort -n file1	Afficher les lignes de file1 triées en ordre croissant, suivant le nombre placé en début de ligne
	Cut -d "/" -f2- file	Eliminer de chaque ligne le nombre et le caractère « / »
	Ls -l grep -c '^-'	Le nombre de fichiers réguliers dans le répertoire courant
	Ls -l grep '^d' wc -l	Le nombre de répertoires dans le répertoire courant
	Ls -l grep '^-' > liste	Créer un fichier « liste » contenant la liste des fichiers réguliers
	Ls -l grep '^l'	Afficher uniquement les fichiers du répertoire courant qui sont des liens symboliques
	Man cp grep -c file	Le nombre de lignes contenant le mot "file" dans le manuel de cp

Cut		
-c	Cut -c5 file	Afficher le 5 ^{ème} caractère de chaque ligne de file
	Cut -c5-10,13 file	Afficher les caractères 5 à 10 et le 13 ^{ème} de chaque ligne

	Cut -c15- file	Afficher tous les caractères à partir du 15 ^{ème}
	Ls -l rep sort -nr +4 -6 head awk '{print \$9}'	Affilcher les 10 plus gros fichiers de rep

Awk		
	Awk 'END {print NR}' file	Afficher le nombre de lignes d'un fichier
-F	Awk -F : '{print "ligne : " NR ; print "1 ^{er} champ : " \$1 ; print "dernier champ : " \$NF}' file	Afficher le premier et le dernier champ de chaque ligne d'un fichier
	Ls -l awk 'NR>1 {print "nom : " \$9 "taille : " \$5 }'	Afficher le nom et la taille des fichiers du répertoire courant
	Ls -lR awk '/^d/ {print \$9}'	Afficher tous les sous-répertoires du répertoire courant
	Awk '/cde ab/ {print \$0}' file	Afficher les lignes qui contiennent ab ou cde
	Awk '/^debut\$/ , /^fin\$/ {print \$0}' file	Afficher toutes les lignes incluses entre 2 lignes égales à « début » et à « fin »
	Ls -l awk -v i=0 'NR>1 {i+=5} END {print i}'	Afficher la taille totale des fichiers affichés par ls
	Awk -F " : " 'BEGIN {OFS=" : "} \$3 >= 500 {tmp=\$1 ; \$1=\$2 ; \$2=tmp} {print \$0}' f1	Remplacer le champ1 par 2 et le champ2 par 1 pour les UID >= 500
	Awk -F " : " 'BEGIN {OFS=" : "} \$5 ~ /^\$/ { \$5=\$1 } {print \$0}' f1	