

## Mémo Commandes UNIX

<b>ls</b>	<b>Afficher les fichiers &lt;chemin&gt;</b>
-m	séparés par virgule
-t	triés par date (le plus récent)
-lu	triés par date de dernier accès et indique cette date (le plus récent)
-F	indiquer les types : f/ (répertoire), f* (exécutable), f@ (lien)
-S	triés par taille (décroissant)
-X	triés par extension
-r	triés par alphabet (inverse)
-R	arbre de fichiers
-l	longue format (droit, appartenance, taille, date/heure modification, nom)
-L	si lien symbolique afficher les informations du fichier référencé
-i	i-node (emplacement, propriétaire, droits, taille, date création, date modif)
-d	ne pas lister le contenu des répertoires
<b>cp</b>	<b>Copier des fichiers dans un répertoire &lt;fichier&gt; &lt;rÉpertoire&gt;</b>
-i	avertit l'existence d'un fichier du même nom et demande de le remplacer
-f	forcer la copie
-l	lien dur
-s	lien symbolique
-p	préserver toutes les informations
-r	copier récursivement
<b>mv</b>	<b>Déplacer fichier &lt;fichier&gt; &lt;rÉpertoire&gt; / Changer nom &lt;f1&gt; &lt;f2&gt;</b>
-b	sauvegarde de fichier avant le déplacer
-i	demande la permission pour chaque fichier & dossier
-u	ne pas remplacer si la date de modification >= de celle du remplaçant
<b>rm</b>	<b>Supprimer fichier &lt;fichier1&gt; &lt;fichier2&gt; ...</b>
-d	supprimer les dossiers vides
-r	supprimer un répertoire et ses sous-répertoires récursivement
-f	supprimer fichiers protégés en écriture et dossier sans confirmation
-i	demande la permission pour chaque fichier & dossier
<b>mkdir</b>	<b>Créer répertoire &lt;rÉpertoire1&gt; &lt;rÉpertoire2&gt; ...</b>
Ø	créer répertoire seulement si parents existent déjà, sinon erreur
-p	créer répertoire et ses parents s'ils n'existent pas
<b>rmdir</b>	<b>Supprimer répertoire &lt;rÉpertoire1&gt; &lt;rÉpertoire2&gt; ...</b>
Ø	supprimer répertoire seulement s'il est vide
-p	supprimer répertoire et sous-répertoires (récursivement) s'ils sont vides
<b>touch</b>	<b>Créer fichier &lt;fichier&gt;</b>
<b>cat</b>	<b>Concaténer fichier</b>
<f1>	concaténer f1 à la fin de stdout
<f1> <f2>	concaténer f2 à la fin de f1
> <f1>	concaténer stdin à la fin de f1 <ctrl+D>
<b>grep</b>	<b>Chercher dans fichier &lt;motif1&gt; ‘ ’ &lt;motif2&gt; &lt;fichier1&gt; &lt;fichier2&gt;</b>
-c	nombre de lignes contenant l'expression
-a	traiter fichier binaire comme fichier texte
-R, -r, -recursive	tous fichiers du répertoire récursivement
-v	lignes ne contenant pas l'expression
-i	ignorer la case
-w	les lignes contenant le mot donné comme motif

<b>sort</b>	<b>Trier lignes fichier &lt;fichier&gt;</b>	
-b	ignorer les blancs au début des lignes	
-d	tri téléphonique : ignorer tous sauf lettres, chiffres et blancs	
-f	minuscules équivalents aux majuscules (non accentuées)	
<b>head / tail</b>	<b>Afficher premières/dernières lignes &lt;fichier&gt;</b>	
-n, --lines N lignes		
-q, --quiet, --silent	ne pas afficher les en-têtes mentionnant les noms de fichiers	
-v, -verbose	toujours afficher les en-têtes mentionnant les noms de fichiers	
--version	précéder par un numéro de version	
<b>diff</b>	<b>Comparer 2 fichiers &lt;source&gt; &lt;cible&gt;</b>	
-a	traiter comme fichiers texte et comparer ligne par ligne	
-b	ignorer les espaces blancs	
-B	ignorer les lignes blanches	
--brief	indiquer si différents seulement	
<b>find</b>	<b>Chercher fichier &lt;répertoire&gt; &lt;critère1&gt; &lt;critère2&gt; ...</b>	
-name	sur nom du fichier	
-perm	sur droits d'accès du fichier	
-links	sur nombre de liens du fichier	
-user	sur propriétaire du fichier	
-group	sur groupe auquel appartient le fichier	
-type	sur type du fichier (d=répertoire, c=caractère, f=fichier normal, l=lien)	
-size	sur taille du fichier en blocs (bloc = 512 octets)	
-atime	sur date de dernier accès en lecture du fichier	
-mtime	sur date de la dernière modification du fichier	
-ctime	sur date de création du fichier	
<b>chmod</b>	<b>Modifier droits &lt;droits&gt; &lt;fichier/répertoire&gt;</b>	
-R	appliquer à tous fichiers et répertoires d'un répertoire	
rwxrwxrwx	3 lettres pour chacun : propriétaire, groupe du propriétaire, autres	
ajouter/retirer/ retirerTous&Ajouter droit	concerné#droit : concerné (u, g, o, a), #(+, -, =), droit(r,w,x) -> concerné#droit : [multipliable séparable par virgule] -> concerné : [multipliable non séparable]	
modifier droits	(3bits)(3bits)(3bits) ⇔ 3 chiffres octaux (ex : 676)	
<b>umask</b>	<b>Changer droits par défaut &lt;valeur&gt;</b>	
par défaut	fichier : 666, répertoire : 777	
Ø	retrancher de droits par défaut la valeur donnée comme argument	
<b>chown</b>	<b>Changer propriétaire du fichier &lt;propriétaire&gt; &lt;fichier&gt;</b>	
<b>chgrp</b>	<b>Changer groupe du fichier &lt;groupe&gt; &lt;fichier&gt;</b>	
<b>cut</b>	<b>Supprimer une partie de chaque ligne &lt;fichier1&gt;</b>	
-d	séparateur des champs	
-f	liste des champs	
-c	sélection sur le rang du caractère	
-b	sélection sur le numéro d'octet	
-s	(avec -f) supprime les lignes vides	

## Commandes simples

<b>id</b>	informations d'id d'utilisateur
<b>whoami</b>	nom d'utilisateur
<b>users</b>	utilisateurs connectés
<b>who</b>	users plus détaillée
<b>passwd</b>	changer mot de passe
<b>groups</b>	groupes auxquels l'utilisateur appartient
<b>newgrp &lt;groupe&gt;</b>	changer le groupe
<b>su &lt;username&gt;</b>	changer l'id de l'utilisateur
<b>lastlog</b>	date de dernière connexion
<b>cd &lt;chemin&gt;</b>	changer de répertoire
<b>pwd</b>	répertoire courant
<b>date</b>	date actuelle
<b>more &lt;fichier&gt;</b>	contenu du fichier (espace : suivant, b : précédent, q : quit)
<b>vi &lt;fichier&gt;</b>	éditer le fichier
<b>man &lt;commande&gt;</b>	manuel de la commande
<b>du</b>	taille de l'arborescence
<b>wc</b>	nombre de lignes, mots et caractères (-l, -w, -c)
<b>paste</b>	regroupe les fichiers ligne par ligne <fichier1> <fichier2> ...
<b>echo "texte"</b>	afficher du texte

## Caractères spéciaux

*	suite de caractères	;	séparateur des commandes
?	un seul caractère	[ ]	choix entre alternatives
espace	séparateur des paramètres	.	répertoire courant
..	répertoire parent	-	répertoire personnel
{, }	ensemble de fichiers	[!]	n'importe quel caractère sauf ces alternatives

## Caractères spéciaux pour la commande grep

.	un seul caractère	^	début du mot
[ ]	choix entre alternatives	\$	fin du mot
[^ ]	n'importe quel caractère sauf ces alternatives		

## Redirection de flux

>	sortie standard vers un fichier en l'écrasant
>>	sortie standard à la fin du fichier
2>	sortie d'erreur vers un fichier en l'écrasant
2>>	sortie d'erreur à la fin du fichier
2>&1	fusionner la sortie d'erreur et la sortie standard
<	lire depuis un fichier
<<	lire depuis le clavier progressivement
	chainer les commandes

## Substitution ou suppression des caractères <ch1> [<ch2>]

Ø	remplacer toutes les occurrences de <ch1> par <ch2>
-s	éliminer les répétitions successives de <ch1>
-d	supprimer toutes les occurrences de <ch1>
-c	complément de <ch1>
[a-z]	segment de a à z \xyz code octal xyz [:upper:] majuscule
[a*n]	a...a n fois [:lower:] minuscule [:alnum:] alphanumérique

## uniq : Eliminer les lignes dupliquées successives <fichier>

-d	affiche seulement les lignes dupliquées
-u	affiche seulement les lignes non dupliquées (par défaut)
-c	nombre d'exemplaires de chaque ligne

## Processus

<b>ps</b>	Afficher les processus en cours d'exécution
-f	format complète
-e	chaque processus
-u ...	appartenant à l'utilisateur ...
-l	format longue
<b>Commandes processus</b>	
<b>top</b>	afficher avec remise à jour les processus en fonctionnement
<b>commande &amp;</b>	lancer le processus de la commande en background
<b>jobs</b>	afficher les processus en background
<b>kill %...</b>	terminer un processus en background
<b>fg %...</b>	mettre une commande en avant plan
<b>bg %...</b>	mettre une commande en arrière-plan
<b>nice -n [p] commande</b>	lancer une commande avec une priorité ∈ [-20(max), 19(min)]
<b>pstree</b>	Afficher les processus en arborescence
-p	afficher les PID des processus entre parenthèses
-h	mettre en surbrillance le processus actuel et ses ancêtres
-a	afficher les arguments de la commande des processus
-c	désactiver le regroupement des processus en une seule ligne
-u ...	appartenant à l'utilisateur ...
<b>kill</b>	Envoyer un signal à un processus <option-signal> <PID>
-l	afficher la liste des signaux
-15 ou -term	terminer un processus (SIGTERM)
-9 ou -kill	tuer un processus
-19	stopper un processus (SIGSTOP)
-18	reprendre l'exécution d'un processus suspendu
-2	interrompre le processus (SIGINT)
<b>pkill</b>	Envoyer un signal à un processus en fonction de propriétés <option-signal> <option-motif> <motif>
-u ...	les processus appartenant à l'utilisateur ...
-n	le processus le plus récent
-x ...	les processus dont le nom est ...
-G ...	les processus appartenant à GUID ...
-t ...	les processus lancés dans le terminal ...
-o ...	l'ancien processus issu de la commande ...
<b>pgrep</b>	Afficher les IDs des processus en fonction de certains critères
-l ...	IDs des processus issu de la commande ...
-t ...	IDs des processus lancés dans le terminal ...
-o ...	ID de l'ancien processus issu de la commande ...
<b>renice</b>	Changer la priorité d'un processus -n [p] <option-motif> <motif>
-P ...	priorité du processus de PID ...
-u ...	priorité des processus appartenant à l'utilisateur ...

## Programmation Shell

Exécuter un script (le droit x est obligatoire)	
<b>bash nomFichier</b>	exécution directe du fichier du script
<b>#!/bin/bash</b>	ajouter à l'entête du fichier et l'exécuter par ./nomFichier

## Variables prédéfinies

HOME	répertoire de login
PATH	répertoire à inspecter pour trouver les commandes tapées par l'utilisateur
TERM	type du terminal
PWD	répertoire courant
USER	utilisateur courant
UID	ID de l'utilisateur courant
GID	ID du groupe de l'utilisateur courant

## Variables de substitution

\$0	nom du script
\$1 à \$9	arguments passés au script
\$#	nombre d'arguments passés au script
\$?	résultat de la commande précédente
\$@	liste des arguments passés au script

**echo** Afficher du texte sur la sortie standard

- Ø avec saut de ligne à la fin
- n sans saut de ligne à la fin

**expr** Evaluer une expression  $\Leftrightarrow \$((...))$

**read** Lire à partir de l'entrée standard <nomVariable> (sans \$)

if	case	while	foreach	for
if [ ... ] ; then ... elif [ ... ] ; then ... else ... fi	case \$var in valeur1) ... ;; valeur2) ... ;; *) ... ;; esac	while [ ... ] do ... done	for i in tableau do ... done tableau : 'seq 0 10' ou \$@ ou chemin/*	for( i=0 ;i<\$var ;i++ ) do ... done

## tests sur les fichiers

[ -e chemin ]	si le fichier existe
[ -d chemin ]	si le fichier est un répertoire
[ -f chemin ]	si le fichier est un fichier ordinaire
[ -L chemin ]	si le fichier est un lien symbolique
[ -r chemin ]	si le fichier est lisible (readable)
[ -w chemin ]	si le fichier est modifiable (writeable)
[ -x chemin ]	si le fichier est exécutable (executable)
[ chemin1 -nt chemin2 ]	si le fichier1 est plus récent que le fichier2
[ chemin1 -ot chemin2 ]	si le fichier1 est plus ancien que le fichier2

## tests sur les chaînes de caractères

[ -z chaine ]	si la chaîne est vide
[ -n chaine ]	si la chaîne est non vide
[ chaine1 = chaine2 ]	si les 2 chaînes sont égales
[ chaine1 != chaine2 ]	si les 2 chaînes sont différentes

## tests sur les nombres

[ nb1 -eq nb2 ]	nb1 == nb2
[ nb1 -ne nb2 ]	nb1 != nb2
[ nb1 -lt nb2 ]	nb1 < nb2
[ nb1 -le nb2 ]	nb1 <= nb2
[ nb1 -gt nb2 ]	nb1 > nb2
[ nb1 -ge nb2 ]	nb1 >= nb2