

# Mémento bash

Valérie Ménissier-Morain

Adaptation de Cécile Braunstein

Version du 11 septembre 2020

Non seulement ce mémento n'a pas de prétention à l'exhaustivité sur le sujet concerné mais une éventuelle exhaustivité sur le sujet serait contre-productive : il a pour but de rappeler l'essentiel et de permettre de retrouver l'information importante parmi la quantité gigantesque d'information disponible.

## Table des matières

### La documentation

#### L'arborescence

Les chemins	1
Raccourcis et méta-caractères	1
Les commandes	1
Visu et modif des infos sur les fichiers	1
Manipulations de fichiers	2
Contenu de flux ou de fichiers (filtres)	2
Comparaison de fichiers	2
Historique de commandes	2
Recherche dans les fichiers et l'arborescence	2
Compression et archivage	3

#### Le Système

Historique de commandes	3
Variables d'environnement	3
Changement d'utilisateur	3
Connexions à distance	3
La mémoire	3

#### Gestion des processus

Visualisation des processus	3
Contrôle des processus	4

#### Gestion des flux

#### Gestionnaire de Version

## 1 La documentation

**man** le manuel au format texte

Options principales :

-k <expression> : affiche la description courte et le nom des pages de manuel comportant l'<expression> (commande apropos)  
<num> : affiche la page de manuel de la section <num>  
-a : affiche la page de manuel pour toutes les sections  
-f : affiche les numéros de sections associées à une commande  
Le manuel utilise la convention suivante : les parties entre [ et ] dans le SYNOPSIS sont optionnelles. Ces caractères ne font pas partie des caractères à taper pour utiliser l'option concernée avec la commande documentée.

**help** : la doc pour les primitives du shell, extrait du **man bash**

De nombreuses commandes intègrent une option -h ou --help qui fournit une aide plus ou moins succincte.

## 2 l'arborescence

### 2.1 Les chemins

Le système de fichier définit l'organisation hiérarchique des fichiers à partir de la racine du système /. Il est alors possible de référencer un fichier de manière absolue, à partir de la racine (/) ou à partir de l'endroit où on se trouve dans la hiérarchie.

Si on se trouve dans le répertoire /home/user, les deux commandes suivantes sont équivalentes.

```
1 user@pc:~$ mv aFile newName
1 user@pc:~$ mv /home/user/aFile /home/user/newName
```

### 2.2 Raccourcis et méta-caractères

~ Répertoire personnel de l'utilisateur.

. Répertoire courant.

.. Répertoire parent.

\* Remplace zéro ou plusieurs caractères.

? Remplace un seul caractère.

Si vous voulez utiliser un méta-caractère ou un espace comme un caractère, ajouter le caractère d'échappement (\) devant.

### 2.3 Les commandes

**pwd** (Print Work Directory)

**cd** (Change Directory)

**cd dirname** se place dans le répertoire dirname

**cd** se place dans le répertoire personnel de l'utilisateur équivalent à **cd ~**

**cd -** se place dans le répertoire précédent

**cd ..** se place dans le répertoire parent

**mkdir** (MaKe DIRectory),

**mkdir -p** crée tous les répertoires intermédiaires nécessaires

**rmdir** (ReMove DIRectory)

**dirname** supprime la dernière partie d'un nom de fichier

**basename** élimine le chemin d'accès (et le suffixe passé en second argument s'il y en a un) du nom de fichier

### 2.4 Visu et modif des infos sur les fichiers

**file** indique le type du fichier

**wc** (Word Count) indique la taille du fichier : nombre de lignes, de mots, de caractères

Options principales :

-c Characters, -w Words, -l Lines

Une option plus occasionnelle : -L affiche le nombre de caractères de la plus longue ligne

**ls** (LiSt, affiche le contenu d'un répertoire),

Options principales :

-a (All) : y compris les noms de fichiers commençant par "." ,

-A : idem sauf "." et "..",

-C (Columns) : en colonnes,

-l : 1 seul fichier par ligne

-d (Directories) : afficher les répertoires comme des fichiers ordinaires

-F (File type) : ajouter un caractère d'indication de type (répertoire, lien symbolique, fichier exécutable, etc.)

-l (Long) : affichage long, voir ci-dessous, y ajouter -h pour avoir les tailles de fichiers plus lisibles

-L (Links) : affiche les infos du fichier cible d'un lien symbolique et non celles de la cible elle-même

-r (Reverted) : afficher en ordre inverse

-R (Recursive) : récursivement

-S (Size) : par ordre de taille décroissante,

-t (Time stamp) : par ordre de date de dernière modification du plus récent au plus ancien

Format de la sortie de **ls -l**

```
% ls -l numbers
-rw-rw-r-- 1 menissie p6ipens 8325 nov 29 2010 numbers
↑      ↑      ↑      ↑      ↑
type de droit utilisateur groupe taille date de nom
fichier fichier physiques octets modification fichier
```

#### Type de fichier

- pour les fichiers ordinaires,  
d pour un répertoire (Directory),  
l pour un lien symbolique

#### Permissions d'accès symboliques

```
rw- rw- r--
user group others
```

r droit de lire (Read)

w droit d'écrire (Write) ou de supprimer

x droit d'exécuter un fichier (eXecute) ou de parcourir un répertoire

- pas le droit correspondant à la position

**lspci** Liste tous les périphériques sur le bus PCI

**lsusb** Liste tous les périphériques connectés en USB

**lshw** affiche les informations détaillées de la configuration matérielle de la machine

Modification des droits

**chmod** (*CHange access MODE*), modification des droits  
Syntaxe : **chmod** <gens><operation><droits> où  
<gens> est une suite (éventuellement vide) de caractères u, g, o et a,  
<operation> est + pour ajouter des droits, - pour en retirer, = pour  
attribuer exactement les droits qui suivent,  
<droits> est une liste (éventuellement vide) de caractères r, w et x.  
Option : -R (*Recursive*)

2.5 Manipulations de fichiers

**rm** (*ReMove*), supprime les fichiers passés en argument

Options principales :  
-f (*Force*) ignore les fichiers inexistant, ne demande pas de confirmation  
-i (*Interactive*) demande confirmation avant chaque effacement  
-r (*Recursively*) récursivement  
**cp** (*CoPy*) copie des fichiers ou des répertoires sources vers la cible

Options principales :  
-r (*Recursively*) récursivement  
-a (*Archive*) copie en préservant le maximum d'informations (droits, propriétaire, liens symboliques, dates, etc)  
**mv** (*MoVe*, renommage et déplacement)  
**ln -s** (*LiNk*, création de lien symbolique)  
**ln -s <source> <cible>**

2.6 Contenu de flux ou de fichiers (filtres)

**cat** (*conCATenate*, affiche une liste de fichiers et l'entrée standard sur la sortie standard),  
Deux utilisations :  
Une liste de fichiers (éventuellement vide) sur la sortie standard  
**cat <fic1> <fic2>...**  
Avec redirection vers un fichier **cat ... > <fic>**

Options principales :  
-n (*Number*) numérote toutes les lignes  
-s (*Squeeze blank*) supprime les lignes vides répétées

**head** affiche le début des fichiers (ou du flux d'entrée)  
Affiche les dix premières lignes par défaut

Option principale :  
-n <num> (*Numbers*) affiche les <num> premières lignes,  
-n -<num> affiche tout sauf les <num> dernières lignes

**tail** affiche la fin des fichiers(ou du flux d'entrée)  
Affiche les dix dernières lignes par défaut

Options principales :  
-n <num> (*Numbers*) affiche les <num> dernières lignes,  
-n +<num> affiche toutes les lignes à partir de la <num>-ième  
-f (*Follow*) affiche les dernières lignes ajoutées au fur et à mesure que le fichier croît  
**less** Affiche un fichier page par page

2.7 Comparaison de fichiers

**diff** et **colordiff** : comparer des fichiers ligne à ligne  
Option principale :  
-u (*Unified*) affiche 3 (paramétrable) lignes de contexte unifié

Les éditeurs de texte dignes de ce nom proposent des fonctionnalités de comparaison : la commande indépendante **gvimdiff** pour **vi**, les fonctions **ediff** et **compare-windows** pour **emacs**.  
Un outil visuel tel que **meld** permet une comparaison et une fusion extrêmement aisées (non disponible par défaut dans les distributions Linux standards).

2.8 Historique de commandes

**history** Affiche l'historique des commandes numéroté (fichier \$ (HOME) / **.history**)  
!! Ré-exécute la dernière commande.  
!-<n> Re-exécute la n<sup>ème</sup> dernière commande.

Accès direct dans le terminal :  
Flèche horizontal haut ou **Ctrl-p** : Parcourt les dernières commandes  
Flèche horizontal bas ou **Ctrl-n** : Revient sur les commandes  
**Ctrl-r** : cherche une commande dans l'historique (**Ctrl-s** si on l'a dépassé).

2.9 Recherche dans les fichiers et l'arborescence

**grep** : recherche dans un fichier ou une liste de fichiers

**grep <expression> <fichier(s)>**  
Principales options :

-e	recherche une Expression Régulière Étendue
-F	recherche rapidement une chaîne de caractères
-e <motif>	protège un motif qui commence par -
-i	ne pas tenir compte des majuscules/minuscules
-v	sélectionner les lignes qui ne correspondent pas au motif
-c	compte les lignes trouvées
-l	afficher seulement le nom du fichier, pas les lignes trouvées
-n	préfixer chaque ligne de son numéro dans le fichier
-q	ne rien afficher sur stdout, pour n'utiliser que le code de retour
-s	ne rien afficher sur stderr en cas de fichiers inexistant ou illisibles

Options supplémentaires de GNU **grep**

--color=<c>   colorier la chaîne trouvée	
avec <c> : never, always ou auto	
-R, -r	récursivement
-o	affiche seulement la partie de la ligne qui correspond au motif
-w	afficher les lignes où le motif correspond à un mot complet
-x	afficher les lignes où le motif correspond à la ligne complète

Voir **grep --help** pour une aide succincte et plus loin pour les [expressions régulières](#)

**find** rechercher des fichiers dans une arborescence et éventuellement appliquer un traitement à la liste de fichiers résultante

**find <path> <critère de sélection> pour simplement afficher la liste des fichiers obtenue**  
**find <path> <critère de sélection> <commande àappliquer> pour appliquer une commande à chacun des fichiers**

<path> est un chemin d'accès à l'arborescence dans laquelle on effectue la recherche  
Expressions fréquentes :

-name <motif> pour choisir la forme des noms de fichiers recherchés, et beaucoup d'autres critères de sélection des fichiers que l'on cherche  
-prune l'expression suivante s'applique aux fichiers qui ne répondent pas au critère sélectionné par l'expression précédente  
-delete supprimer les fichiers sélectionnés  
-exec <commande> \; appliquer la <commande> aux fichiers sélectionnés, la position du fichier dans l'appel de commande est notifiée par {}, par exemple **find . -exec grep toto {} \;** recherche **toto** dans toute l'arborescence courante

**locate** : recherche de fichiers à partir d'une partie de leur nom  
Principales options :

-i : *case Insensitive* pour ne pas tenir compte des majuscules/minuscules  
-r <motif> (*Regexp*) pour rechercher un <motif>

## 2.10 Compression et archivage

**gzip**, **bzip2**, **xz** : utilitaires de compression

Options principales communes :

-d : décompression  
-c : décompression sur la sortie standard et non pas dans un fichier

**tar** (Tape ARchive) : gestion d'archive

**tar** -cf <archive> <liste des fichiers à archiver>  
**tar** -xf <archive>

Options principales :

-f <archive> (File) fournit le nom de l'<archive>  
-c (Create) crée une nouvelle archive  
-x (eXtract) extrait les fichiers contenus dans l'archive  
-t (Tree) affiche la liste des fichiers contenus dans l'archive  
-C <dir> (Change directory), à placer avant toute autre instruction, effectue l'opération suivante à partir du répertoire <dir>  
-I <compresseur> compresser avec <compresseur> une commande de compression qui accepte l'option -d pour décompresser  
-z : compression avec **gzip**  
-j : compression avec **bzip2**  
-J : compression avec **xz**  
-a : le suffixe du nom de l'archive (.gz, .bz2, .xz) à la création indique que l'archive sera compressée et quel logiciel de compression correspondante sera utilisée  
-p (preserve Permissions) : extrait les informations sur les permissions de fichiers  
-v (Verbose) : affiche la liste détaillée des fichiers traités

## 3 Le Système

**uname** Affiche les informations du système

-s affiche le nom du noyau  
-n affiche le nom de la machine  
-m affiche le type de processeur  
-o affiche le nom de l'OS  
-a affiche tout

**shutdown** Éteint ou reboot de la machine

-r reboot  
-p éteindre  
-t temps en seconde  
-c annuler  
now maintenant

### 3.1 Historique de commandes

**history** Affiche l'historique des commandes numéroté (fichier \$ (HOME) / **.history**)

**!!** Ré-exécute la dernière commande.

!**<n>** Re-exécute la n<sup>ème</sup> dernière commande.

Accès direct dans le terminal :

Flèche horizontale haut ou Ctrl-p : Parcourt les dernières commandes  
Flèche horizontale bas ou Ctrl-n : Revient sur les commandes  
Ctrl-r : cherche une commande dans l'historique (Ctrl-s si on l'a dépassé).

## 3.2 Variables d'environnement

Ce sont des variables utilisées par le système d'exploitation et les programmes. Elles servent à fournir ou communiquer des informations.

HOME (chemin du répertoire personnel de l'utilisateur)  
PATH (contient tous les chemins vers des exécutables)  
LANG (langue utilisée par défaut)  
EDITOR (éditeur de texte par défaut)

**printenv** Visualise la variable équivalent à **echo** \$VAR

**setenv** ou VARIABLE=valeur Création/modification

**export** rend exportable la variable. Si elle ne l'est pas elle reste local et n'est pas recopiée par les processus fils

**env** Affiche toutes les variables d'environnement et leur valeur

### 3.3 Changement d'utilisateur

**sudo** (substitute user-do) Exécute une commande avec d'autres privilèges que les siens.

-s importe les var. d'environnement

**su** change d'utilisateur

### 3.4 Connexions à distance

**ssh** : connexion à un ordinateur distant.

```
user@pc:~$ ssh [-Y] login@mach_dist
```

-Y export de l'interface graphique

**scp** : copie de ou sur un ordinateur distant.

```
user@pc:~$ scp local_file login@mach_dist:/path  
user@pc:~$ scp login@mach_dist:/path/aFile /local_path/
```

**sshfs** : Monter un répertoire distant en local (logiciel fuse)

```
user@pc:~$ sshfs login@mach_dist:/path rep_local
```

Le répertoire doit ensuite être démonté

```
user@pc:~$ fusermount -u rep_local
```

## 3.5 La mémoire

**du** (Disk Usage) affiche l'espace du disque utilisé par les fichiers et les répertoires

-s Affiche le total sans lister les fichiers  
-h Ajoute l'unité (K,M,G)  
-a Pour tous les fichiers et les répertoires

**df** (Disk Free) affiche la quantité d'espace utilisé par les systèmes de fichiers

-a Pour tous les systèmes fichiers y compris ceux de 0 blocs  
-h Ajoute l'unité (K,M,G)  
-T affiche le type de système de fichier

## 4 Gestion des processus

### 4.1 Visualisation des processus

**ps** (Process Status) : affiche par défaut pour les processus de l'utilisateur dans le terminal courant : le pid (numéro de processus), le tty (nom du terminal), le temps d'exécution et le nom.

Options principales

Sélection des processus

-A (All) ou -e : tous les processus  
-C <commande> : affiche l'information pour la commande <commande>

Affichage

u (User) : plus d'infos orienté utilisateur  
f (Filiation) : arborescence des processus  
w (Wide) : ligne plus longue (ww : aussi long que nécessaire)  
o <champs> : choix des <champs> affichés  
--sort <champ> : tri selon le <champ>

États des processus le champ STAT de l'affichage

R : en cours d'exécution  
S : dormant interruptible  
D, Z : mauvais signe !

**top** : liste dynamique des tâches

Commandes interactives principales :

? ou h résumé des commandes  
d (Delay) modifier le délai de rafraîchissement (3 secondes par défaut)  
k (Kill) envoyer au processus dont on précise ensuite le pid le signal choisi ensuite  
q (Quit)

Le processus que l'on cherche à tuer parce qu'il consomme trop de mémoire ou de temps cpu se trouve dans les premières lignes

**htop** : liste dynamique des tâches

Plus ergonomique que top :

commandes interactives en nombre limité présentées sur la dernière ligne, déplacement horizontal et vertical dans la liste affichée, choix du processus en se déplaçant dans la liste (pas besoin de trouver le pid pour le tuer), nom des commandes non tronqués, affichage plus stable, se souvient des préférences d'affichage

Principales commandes :

” ” pour marquer un processus  
U (*Unmark*) pour retirer les marques  
M (*Memory*) pour trier selon l'occupation en mémoire  
P (*Processor usage*) pour trier par consommation CPU  
T (*Time*) pour trier par temps d'exécution  
F (*Follow*) pour que l'affichage reste focalisé sur le processus sélectionné  
H (*Hide user threads*) masque ou affiche les processus légers de l'utilisateur

## 4.2 Contrôle des processus

**jobs** : affiche les processus lancés en tâche de fond par le shell courant

**bg** : place en tâche de fond un processus en avant-plan, pour lancer un processus directement en tâche de fond  
<commande> &

**fg** : fait passer en avant-plan une tâche de fond,

**kill** : envoie un signal à un processus

Le processus est identifié soit par son *pid* visible avec **ps** soit par %<num> où <num> est son numéro affiché par **jobs**

Options principales :

-s <signal> envoie le signal <signal> à son argument  
-l pour visualiser la liste des signaux (disponible aussi par **man** 7 signal)

**killall** : comme **kill** mais à partir du nom de la commande

Signaux principaux :

HUP (*Hang Up*, 1) quitter le processus en lui envoyant un signal de déconnexion  
INT (*Interrupt*, 2) interrompre un processus depuis le clavier (^c)  
KILL (9) quitter le processus  
TERM (*TERMinate*, 15) terminer le processus le plus normalement possible (envoyé par défaut par **kill**, **top** et **htop** pour tuer un processus)

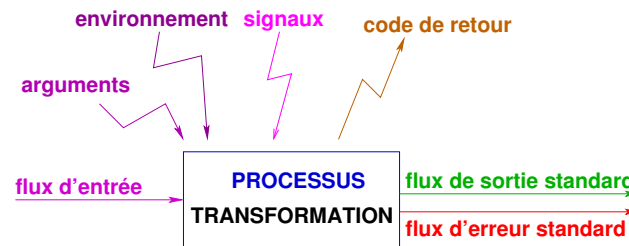
STOP (*STOP*, 19) : stopper le processus  
STP (*SToP*, 20) : stopper le processus (^z)  
CONT (*CONTinue*, 18) : relancer un processus stoppé (**bg**, **fg**)

On ne peut ni capturer ni ignorer les signaux KILL et STOP (versions plus impératives de INT et STP).

**nohup** exécute une commande en la rendant insensible aux déconnexions et en redirigeant l'entrée et les sorties standards si nécessaire, c'est-à-dire si elles ne sont pas déjà redirigées :

L'entrée standard lit alors sur /dev/null  
La sortie standard est ajoutée à **nohup.out**  
La sortie d'erreur standard est redirigée vers la sortie standard  
Utilisation classique : en redirigeant la sortie standard dans <fic> par  
**nohup** <commande> > <fic>

## 5 Gestion des flux



Syntaxe d'une redirection :

<op> <mot> ou <d><op> <mot> où <d> est un descripteur sous forme d'entier avec

flux d'entrée standard (stdin) : numéro 0  
flux de sortie standard (stdout) : numéro 1, > équivalent à 1>,  
flux de sortie d'erreur (stderr) : numéro 2,

<op> est une opération de redirection <, <<, <<-, <<<, >,>>

Redirections usuelles ( <, >, >> )

<commande> < <fichier> redirection de l'entrée standard de la <commande> (ouverture de <fichier> en lecture)

<commande> > <fichier> (resp. <commande> 2> <fichier>) redirection de la sortie standard (resp. d'erreur) de la <commande> (ouverture de <fichier> en écriture avec écrasement du contenu précédent)

<commande> >> <fichier> (resp. <commande> 2>> <fichier>) redirection de la sortie standard (resp. d'erreur) de la <commande> (ouverture de <fichier> en écriture avec ajout au contenu précédent)

Documents en ligne (*here-document*, << et <<-), chaîne en ligne <<<

<commande> <<EOF  
<texte>  
EOF  
<commande> <<< <chaîne>  
avec EOF seul sur une ligne dans un script, <texte> un ensemble de lignes de texte avec évaluation des variables  
<<- idem mais les blancs au début des lignes de <texte> sont supprimés, cela permet de respecter l'indentation d'un script sans polluer le texte produit  
<commande> lit la <chaîne> évaluée sur son entrée standard.

Flux de commandes qui s'enchaînent (cascade, tube ou *pipe*, |)

Le flux de sortie standard d'une commande est le flux d'entrée d'une autre commande : <commande1> | <commande2>

Duplication d'un descripteur de sortie >&

<d1> >&<d2> l'écriture sur le descripteur <d1> écrit sur le descripteur <d2>  
Écrire sur la sortie d'erreur **echo** <message> 1>&2  
Rediriger les deux flux de sortie dans un même fichier  
<commande> > <log> 2>&1

## 6 Gestionnaire de Version

**git** Commande pour le gestionnaire de version Git

**git init** : initialise un répertoire existant comme un dépôt Git.

**git clone** <url> : récupère un dépôt Git à partir de l'url.

**git status** : montre l'état du répertoire : les fichiers modifier, prêts pour le prochain commit/

**git add** : ajout un fichier pour le prochain commit.

**git reset** : retire les fichiers pour le prochain commit.

**git commit -m** '<un message>' : *commite* les changements préalablement ajouter pour créer un nouvel instantané du dépôt.

**git push** : transmet les *commit* locaux à un dépôt distant

**git pull** : récupère les modifications d'un dépôt distant.

**git rm** <file> : efface un fichier d'un projet .