

[Download as PDF](#)

# Memento Unix

---

Tout système Unix (Linux, BSD/Mac OS X, System V...) foisonne de commandes. Sans prétendre à l'exhaustivité, nous espérons ici résumer celles qui vous sont à la fois les plus utiles au quotidien et les plus génériques possibles. Les commandes spécifiques à un Unix particulier sont précisées, nous fournissons également le cas échéant les différences entre les shells Bourn/Korn **sh** **bash** **zsh** **ksh** et les C shells **cs**h et **ts**h.

## Aide

---

### Obtenir de l'aide sur un sujet

```
$ man [section] sujet
```

### Lister les pages man parlant d'un sujet

```
$ man -k sujet  
$ apropos sujet
```

### Afficher la page d'un manuel contenue dans le fichier pageman.X

```
$ groff -Tascii -man pageman.X | less
```

### Obtenir de l'aide sur un sujet (page info)

```
$ info sujet
```

Les pages info sont presque exclusivement utilisées par les programmes de la FSF (Free Software Foundation) : gcc, ar, nm, objdump, ranlib, ld, libtool, autoconf, automake... Ils sont aussi dotés de pages man qui sont souvent obsolètes.

### Référence des sections du man

1. commandes
2. appels système
3. appels de bibliothèques
4. interface noyau et drivers
5. fichiers de configuration
6. jeux
7. standards
8. commandes d'administration
9. documentation du noyau

## Confort et shells

---

## Pour identifier l'interpréteur de commandes actif

```
$ echo $SHELL
```

## Pour changer d'interpréteur

```
$ chsh
```

## Les différents shells

Shell	Commentaires
/bin/sh	Le shell POSIX, anciennement Bourne shell. Peu tourné vers l'utilisation interactive, il est le seul à être standardisé. À préférer pour les scripts. <b>Attention:</b> sur les Unix tels que Linux ou Mac OS X, /bin/sh est un lien vers /bin/bash.
/bin/ksh	Très répandu, le Korn shell est un Bourne shell amélioré pour l'utilisation interactive.
/bin/csh	Le C-shell est un shell utilisant une syntaxe proche du langage C.
bash	Shell standard sous Linux, le Bourne-again shell est un Bourne shell très amélioré au niveau de l'utilisation interactive.
tcsh	Un C-shell amélioré.
zsh	Shell peu répandu mais empruntant à bash et tcsh pour pousser plus loin les fonctionnalités et l'ergonomie interactive.

**Attention:** Sur certain Unix, lorsque l'on configure pour root un shell qui n'est pas sur la partition racine, on rend le monde mono-utilisateur inutilisable.

## Raccourcis clavier

Ils sont disponibles sous **bash**, **csh** et **tcsh** par défaut, mais il ne le sont pas sous **sh**. Sous **ksh**, il faut activer ces raccourcis clavier avec

```
$ set -o emacs
```

**Remarque:** Les touches fléchées permettent également de déplacer le curseur et de naviguer entre les commandes précédentes et suivantes si le terminal est bien configuré.

Raccourcis	Commentaires
Ctrl-P	Commande précédente
Ctrl-N	Commande suivante
Ctrl-A	Début de ligne
Ctrl-E	Fin de ligne
Ctrl-K	Efface la ligne à partir du curseur
Ctrl-U	Efface la ligne jusqu'au curseur en <b>bash</b> ou toute la ligne <b>csh</b> , <b>tcsh</b> et <b>ksh</b> .
Ctrl-B	Déplace le curseur vers le début de la ligne
Ctrl-F	Déplace le curseur vers la fin de la ligne

## Les variables d'environnement

Le contenu entre (" ") est interprété par le shell, le contenu par apostrophes ( ' ') ne l'est pas. Dans tous les cas, si l'élément considéré ne contient pas de caractère spéciaux, guillemets et apostrophes sont

facultatifs.

Actions	Commandes
Afficher une variable VAR	echo \$VAR
Affiche toutes les variables	set
Positionne la variable VAR à la valeur val	VAR='val' ; export VAR sous sh — export VAR='val' sous bash, sh et ksh
Positionne la variable d'environnement VAR à la valeur val	set VAR='val' sous csh
Définit une variables pour une commande commande	VAR='val' commande

## Variables fréquemment utilisées

Variables	Commentaires
SHELL	Nom de l'interpréteur de commandes
PATH	Chemin de recherche des exécutables
LD_LIBRARY_PATH	Chemin de recherche des bibliothèques (sauf sous MacOS X "DYLD_LIBRARY_PATH")
MANPATH	Chemin de recherche des pages de man
HOME	Répertoire maison
LANG	Localisation (ex : fr_FR.ISO8859-1)
LC_CTYPE	Localisation de la classification des caractères et de la casse
LC_ALL	Écrase les définitions d'autres paramètres régionaux LANG, LC_..
DISPLAY	Serveur X, i.e., graphique, vers lequel rediriger l'affichage, e.g., DISPLAY=0 pour local
LESSCHARSET	Jeu de caractères pour la commande less. Valeurs possibles: ascii, latin1, utf-8, dos, ebcdic, IBM-1047, koi8-r, next
IFS	Séparateur de champs ou de mots lors de l'interprétation d'une chaîne (sh, ksh, bash). NB: les espaces (tabulations, retour chariot, caractère espace) sont gérés différemment des autres caractères. Une série de 3 tabulations sera interprétée comme un seul séparateur tandis que ':::' sera interprété comme trois séparateurs de champs.
LOGNAME	Login de l'utilisateur
PWD	Le répertoire courant
PS1	Invite de l'interpréteur de commande (bash) — \u usager — \h nom de la machine — \W dernier niveau du répertoire courant — \w répertoire courant — \A heure au format HH:MM (24heures) — \\$ symbole \$. NB: Valable aussi pour sh et ksh mais sans les séquences d'échappement.
prompt	Invite de l'interpréteur de commande csh — %n usager — %m nom de la machine — %C répertoire courant — %CX X dernier niveau du répertoire courant — %T heure au format HH:MM (24heures)
EDITOR	Éditeur de texte par défaut

## Naviguer dans les répertoires

Commandes	Commentaires
pwd	Affiche le répertoire courant
cd rep	Déplacement dans le répertoire rep
cd	Déplacement dans le répertoire home, équivalent à cd ~
cd ..	Déplacement dans le répertoire parent
ls [rep]	Liste les fichiers du répertoire rep (si un répertoire n'est pas spécifié, les fichiers du répertoire courant sont listés)

<code>ls</code> <code>[rep]</code>	<code>-a</code>	Liste aussi les fichiers cachés
<code>ls</code> <code>[rep]</code>	<code>-l</code>	Liste les fichiers avec leur droits d'accès

## Gérer les droits d'accès

### Conventions pour les droit d'accès

Le 1er caractère correspond au type d'objet (fichier, répertoire, etc.). Les suivants se lisent par groupes de 3 et définissent respectivement les droits accordés:

1. au propriétaire
2. au groupe propriétaire du fichier et
3. aux autres usagers.

### Types d'objets

Symboles	Types
–	fichier
d	répertoire
l	lien symbolique
s	socket
p	FIFO
c	device (mode caractère)
b	device (mode bloc)

### Types de droits accordés

Symboles	Droits accordés
–	pas d'accès
r	lecture
w	écriture
x	exécution (fichier), parcourir (répertoire)
s	bit set-UID (utilisateur), bit set-GID (groupe)
t	sticky bit: impossible d'effacer les fichiers des autres
S	bit set-UID sans droit en exécution
T	sticky bit sans droit de parcourir

### Changer le propriétaire d'un fichier

```
$ chown usager fichier
```

### Changer le groupe d'un fichier

```
$ chgrp groupe fichier
```

## Changer les droits d'accès d'un fichier

```
$ chmod mode fichier
```

ou mode est:

- soit sous la forme groupe(s) + droit(s) (ajout de droits)
- soit sous la forme groupe(s) – droit(s) (suppression de droits) avec groupe : usager **U**, groupe **g** et autres **O**.
- soit sous la forme octale **SUGO** ou **S = 1** (sticky-bit) ou **2** (set-GID) ou **4** (set-UID) et **U**, **G** et **O** peuvent chacun prendre pour valeur **1** (exécutable) ou **2** (droit en écriture) ou **4** (droit en lecture). On additionne les modes par chiffre pour obtenir un mode de quatre chiffre.

Exemples: changer les droits sous la forme groupe + droit

- Permet à tous de lister et de parcourir un répertoire

```
$ chmod ugo+rx repertoire
```

- Retire tous les droits aux utilisateurs autres que le propriétaire du fichier.

```
$ chmod go-rwx fichier
```

Exemples: changer les droits sous la forme octal **SUGO**

- L'utilisateur courant et le groupe peuvent parcourir le répertoire et le lire

```
$ chmod 1750 repertoire
```

- Le fichier est accessible en lecture et en écriture au groupe et à l'utilisateur propriétaire du fichier

```
$ chmod 660 fichier
```

## Manipuler les fichiers

Commandes	Commentaires
<code>mv source cible</code>	Déplace et/ou renomme un fichier
<code>cp source cible</code>	Copie un fichier
<code>cp -R source cible</code>	Copie un répertoire. <b>Attention:</b> les droits d'accès peuvent être perdus. Pour éviter cela utiliser l'option <code>-p</code> ou utiliser <code>tar</code>
<code>ln source lien</code>	Lien fort de source vers lien
<code>ln -s source lien</code>	Lien symbolique de source vers lien
<code>mkdir repertoire</code>	Crée un répertoire
<code>rmdir</code>	Supprime un répertoire vide
<code>du repertoire -ks</code>	Affiche la taille du répertoire en Ko

rm fichier	Supprime un fichier ( <b>Irréversible!</b> )
rm -f fichier	Supprime un fichier protégé en écriture
rm -R répertoire	Supprime un répertoire
touch fichier	Met à jour la date de modification d'un fichier

## Rechercher un fichier

Commandes	Commentaires
locate motif	Recherche des fichiers dont le nom correspond au motif
updatedb	Met à jour la base de données de locate ( <b>linux</b> )
/usr/libexec/locate.updatedb	Met à jour la base de données de locate ( <b>BSD, MacOS X</b> )
find chemin options	Recherche les fichiers correspondant aux options dans chemin

### Principales options de la commande "find"

- Recherche à partir du répertoire courant les fichiers dans le nom respectif le motif

```
$ find . -name motif
```

- Recherche à partir du répertoire courant les fichiers du type spécifié : **f** pour les fichiers réguliers, **d** pour les répertoires et **l** pour les liens symboliques

```
$ find . -type type
```

- Exécute la commande **cmd** sur tous les fichiers trouvés. L'argument **{}** désigne le nom du fichier trouvé par **find**

```
$ find . -exec cmd \;
```

- Affiche les fichiers trouvés (**System V** uniquement, implicite pour **linux, MacOS X** et **BSD**)

```
$ find . -print
```

### Exemples

- Recherche dans le répertoire courant tous les fichiers dont l'extension est **html**

```
$ find . -name '*.html'
```

- Supprime à partir du répertoire courant tous les fichiers dont l'extension est **png**

```
$ find . -name '*.png' -exec rm {} \;
```

## Redirection des flux

Commandes	Commentaires
cmd > fichier	Redirige la sortie standard de la commande <b>cmd</b> dans le fichier spécifié

cmd > > fichier	Écrit à la fin du fichier au lieu de l'écraser
cmd < fichier	Utilise le fichier spécifié comme entrée de la commande cmd
cmd1   cmd2	Utilise la sortie de la commande cmd1 comme entrée de la commande cmd2
cmd < < "chaîne"	Utilise l'entrée standard pour la commande cmd jusqu'à que la chaîne spécifiée soit rencontrée
cmd 2>&1 > fichier	Redirige la sortie standard et les erreurs dans le fichier spécifié ( <b>sh</b> , <b>bash</b> , <b>ksh</b> )
cmd >& fichier	Redirige la sortie standard et les erreurs dans le fichier spécifié ( <b>cs</b> , <b>tcsh</b> )
cmd > fichier1 2> fichier2	Redirige la sortie standard dans le fichier1 et les erreurs dans le fichier2 ( <b>sh</b> , <b>bash</b> , <b>ksh</b> )
(cmd > fichier1) >& fichier2	Redirige la sortie standard dans le fichier1 et les erreurs dans le fichier2 ( <b>cs</b> , <b>tcsh</b> )

### Exemple

- Liste tous les fichiers du répertoire courant avec leurs droits d'accès dans le fichier `liste_fichiers`

```
$ ls -al > liste_fichiers
```

## Comparer et trier des fichiers

- Affiche les différences entre les deux fichiers texte spécifiés

```
$ diff fichier1 fichier2
```

- Affiche les différences entre les deux fichiers spécifiés au format `patch`

```
$ diff -u fichier1 fichier2
```

- Compare et affiche un contexte de deux lignes pour chaque différence constatée

```
$ diff -C 2 fichier1 fichier2
```

- Compare deux fichiers binaires spécifiés

```
$ cmp fichier1 fichier2
```

- Compare le premier fichier à partir du 13ème octet et le second fichiers à partir du 17ème

```
$ cmp fichier1 fichier2 13 17
```

- Trie les lignes d'un fichier. Options:
  - `-n` Tri numérique
  - `-k n` Tri sur le nème champs de la ligne

```
$ sort fichier
```

- Supprime les doublons de lignes dans un fichier trié (l'option `-C` permet d'afficher le nombre d'occurrences devant chaque ligne)

```
$ uniq fichier
```

## Exemples

- Stock dans un fichier `main.patch` les différences entre deux fichiers sources

```
$ diff -u logiciel2.0/main.c logiciel1.0/main.c > main.patch
```

- Affiche les différences entre deux fichiers sources en précisant 2 lignes de contexte

```
$ diff -C 2 logiciel2.0/main.c logiciel1.0/main.c
```

## Afficher le contenu de fichiers

Commandes	Commentaires
<code>wc fichier</code>	Compte le nombre de lignes, de mots, d'octets d'un fichier spécifié
<code>cat fichier1 fichier2</code>	Concatène les fichiers spécifiés
<code>more fichier</code>	Affiche un fichier page après page: — espace pour passer à la page suivante — entrée pour passer à la ligne suivante — U pour remonter — q pour quitter.
<code>less fichier</code>	Affiche un fichier page après page et permet d'y naviguer avec les flèches Page prec et Page Suiv.
<code>head -n 3 fichier</code>	Affiche les 3 premières lignes du fichier spécifié
<code>tail -n 3 fichier</code>	Affiche les 3 dernières lignes du fichier spécifié
<code>tail -f fichier</code>	Affiche la fin du fichier au fur et à mesure qu'il est rempli
<code>echo 'chaîne' &gt; fichier</code>	Crée le fichier avec le nom spécifié contenant la chaîne en argument
<code>echo 'chaîne' &gt; &gt; fichier</code>	Crée la chaîne en argument à la fin du fichier après un saut de ligne
<code>cat - &gt; fichier « EOF</code>	Écrit ligne à ligne dans le fichier jusqu'à la ligne EOF (End of file)

## Exemples d'utilisation

- Affiche le fichier `/etc/resolv.conf`

```
$ cat /etc/resolv.conf
```

- Liste tous les fichiers du répertoire courant et les affiche page après page

```
$ ls -as | more
```

- Affiche les 3 premières lignes du fichier `todoList`

```
$ head -n 3 todoList
```

- Affiche les 3 dernières lignes du fichier de log `/var/log/message`



```
$ tail -n 3 /var/log/message'
```

- Affiche la fin du fichier de log d'erreur du serveur Web apache `/var/log/apache/error.log` au fur et à mesure qu'il est rempli

```
$ tail -f /var/log/apacheerror.log
```

- Écrit la ligne `licorne` dans le fichier `/etc/hostname` autrement dit changement le nom réseaux de la machine

```
$ echo 'licorne' > /etc/hostname
```

- Écrit ligne à ligne dans le fichier `list_de_courses-0806.txt` jusqu'à ligne EOF

```
$ cat - > list_de_courses-0806.txt << "EOF"
```

## Recherche de chaînes avec grep

La commande `grep` recherche un motif dans un ensemble de chemins spécifiés et affiche les lignes correspondantes:

```
$ grep motif chemins
```

- Exemple: recherche tous les fichiers du répertoire `"Mail/inbox/"` avec un nom contenant la chaîne de caractères `"recette de cassoulet"`:

```
$ grep "recette de cassoulet" Mail/inbox/*
```

Options	Commentaires
<code>grep -A N</code>	Affiche la ligne et les <code>N</code> lignes suivantes ( <b>Linux</b> , <b>BSD</b> , <b>MacOs X</b> )
<code>grep -B N</code>	Affiche la ligne et les <code>N</code> lignes précédentes ( <b>Linux</b> , <b>BSD</b> , <b>MacOs X</b> )
<code>grep -i</code>	Le motif est insensible à la casse
<code>grep -P</code>	Le motif est donné comme une expression régulière Perl ( <b>Linux</b> , <b>BSD</b> , <b>MacOs X</b> )
<code>grep -R</code>	Le motif est recherché récursivement dans les sous-répertoires ( <b>Linux</b> , <b>BSD</b> , <b>MacOs X</b> )
<code>grep -E</code>	Le motif est une expression régulière étendue (voir ci-dessous)

**Remarque:** Lorsque la commande `grep -R` n'est pas disponible, on peut effectuer la recherche récursivement une chaîne avec la commande `find`:

```
$ find chemin -exec grep motif {} \; - print
```

### Les expressions régulières étendues

Elles consistent en des chaînes de caractères modèle auxquelles comparer d'autres chaînes. Dans une expression régulière, certains caractères revêtent un sens particulier:

Caractères	Significations
<code>.</code>	n'importe quel caractère

[0-9]	un chiffre
[A-Z]	une majuscule entre A et Z
[a-z]	une minuscule entre a et z
[A-Z0-9]	une majuscule entre A et Z ou un chiffre
[a,g,h,i]	l'un des caractère a, g, h ou i

À ces atomes peuvent être ajoutés des localisateurs:

Localisateurs	Significations
^E	un E en début de ligne
toto\$	la chaîne "toto" en fin de ligne

et des quantifieurs:

Quantifieurs	Significations
E*	un nombre quelconque et éventuellement aucun E
[0-9]+	un nombre quelconque et au moins un chiffre
[xyz]?	rien ou une lettre parmi X, y ou z
.{3}	3 caractères n'importe lesquels
Z{2,}	au moins 2 caractères Z
[A-Z]{,6}	au plus 6 majuscules
[a-z]{3,5}	entre 3 et 5 minuscules

Enfin on peut factoriser des expressions et indiquer des alternatives:

Factorisateurs	Significations
(toto){2}	la chaîne "toto" répétée deux fois
(toto   tata)\$	la chaîne "toto" ou "tata" en fin de ligne

Le caractère « \ » permet de faire perdre son sens particulier (« échapper ») à l'un des caractères spéciaux.

## Remplacer des chaînes

### La commande sed

Remplace l'expression régulière **expr** par la chaîne spécifiée dans un fichier

```
$ sed 's/expr/chaîne/' fichier
```

Remplace directement l'expression régulière **expr** par la chaîne spécifiée dans un fichier et enregistre l'ancien fichier avec le nom **fichiersuff** (Linux)

```
$ sed -i suff 's/expr/chaîne/' fichier
```

Remplace plusieurs occurrences par ligne

```
$ sed 's/expr/chaîne/g' fichier
```

Remplace l'expression régulière `expr` par une chaîne: `\1` est remplacé par ce qui correspond à la partie de l'expression régulière entre `\(` et `\)`

```
$ sed 's/\(groupe\)expr/chaîne\1/' fichier
```

**Remarque:** Il peut y avoir plusieurs groupes entre parenthèse dans le premier membre qui se substitue à `\1` `\2...` `\9` dans le second membre.

**Exemple:** remplace tous les noms de fichier se terminant par `.html` par des noms de fichier se terminant pas `.fr.html`

```
$ sed 's/\(.*\)\.html/\1.fr.html/' fichier
```

## La commande tr

Remplacement sur le flux d'entrée standard: les `a` sont transformé en `d`, les `b` en `e` et les `c` en `f`.

```
$ tr 'abc' 'def'
```

**Remarque:** On peut utiliser des classes de caractères telles que `'[A-Z]'`.

Exemples (La conversion de fichier)

- Conversion de fichier LF Unix  $\Rightarrow$  CR MacOS

```
$ tr '\n' '\r' < in > out
```

- Conversion de fichier LF Unix  $\Rightarrow$  CRLF MS-DOS

```
$ tr '\n' '\r\n' < in > out
```

- Conversion majuscules  $\Rightarrow$  minuscule

```
$ tr '[A-Z]' '[a-z]' < in > out
```

## Éditer des fichiers

### La commande cut

La commande `CUT` affiche les champs d'un fichier (séparés par des tabulations, index commençant à 1):

- `X,Y`: champs `X` et `Y`
- `X-Y`: champs entre `X` et `Y` inclus
- `-d`: délimiteur.

```
$ cut -f champs fichier
```

**Exemple:** Extrait la deuxième et la cinquième colonne d'un fichier nommé `data.csv` (séparation des champs par le point-virgule).

```
$ cut -f 2,5 -d ';' data.csv
```

## La commande paste

La commande **paste** affiche des lignes contenant la ligne correspondante de chaque fichier séparées par des tabulations:

- **-d**: délimiteur

```
$ paste f1 f2 ...
```

**Exemple:** Si les fichiers **f1**, **f2** et **f3** sont des fichiers numériques, affiche des lignes de nombres sous la forme **x1:x2:x3**.

```
$ paste -d ':' f1 f2 f3
```

## La commande join

La commande **join** permet de fusionner deux fichiers ayant un champ commun (jointure):

- **-1 champ**: champ de jointure dans le fichier 1
- **-2 champ**: champ de jointure dans le fichier 2

```
$ joint fichier1 fichier2
```

## La commande vi

La commande **Vi** permet d'éditer un fichier. Pour plus d'information vous pouvez consulter le [memento Vi](#).

```
$ vi fichier
```

Vi dispose de deux modes: édition et commande, qui est le mode au lancement. En mode édition, **ESC** permet de passer en mode commande. En mode commande:

- **:w** enregistre
- **:q** quitte
- **:wq** enregistre et quitte
- **:q!** quitte sans enregistrer

La manipulation du curseur s'effectue avec les touches fléchées si le terminal est bien configuré ou avec les touches:

- **h** pour se diriger à gauche
- **j** pour se diriger en bas
- **k** pour se diriger en haut
- **l** pour se diriger à droite

Les commandes d'édition les plus courantes:

- **X** efface un caractère

- **r** remplace un caractère
- **dd** coupe la ligne courante
- **yy** copie la ligne courante
- **u** annuler
- **3yy** copie 3 lignes à partir de la ligne courante
- **p** colle la dernière ligne coupée ou copier après la ligne courante
- **1,\$s/foo/bar/g** remplace **foo** par **bar** (même syntaxe que [la commande sed](#))
- **!** execute une commande externe sur le fichier, e.g.,

```
: 1,$!tr 'A-Z' 'a-z'
```

## Réseaux

Commandes	Commentaires
ping [-n] machine	Envoie un ping à la machine spécifiée (-n sans DNS)
tracert [-n] machine	Trace la route d'un paquet de données vers machine spécifiée (-n sans DNS)
netstat [-n]	Affiche l'usage du réseaux par les processus
netstat [-n] -a	Idem avec affiche des processus serveur
fuser, fstat, lsof	Liste détaillée de l'usage des fichiers et du réseau
ifconfig -a	Affiche la configuration des interfaces réseaux
ifconfig interface IP masque	Configure une interface réseau
route [-n] show	Affiche la table de routage (-n sans DNS et sans l'option show pour les systèmes <b>Linux</b> )
route [-n] add route [gw] passerelle	Ajoute une entrée de routage (gw uniquement pour <b>Linux</b> )
route add default [gw] passerelle	Ajoute une entrée de routage par défaut (gw uniquement pour <b>Linux</b> )
route delete default	Efface la route par défaut
hostname	Affiche et configure le nom réseau de la machine
/etc/resolv.conf	Fichier de configuration de la résolution DNS

### Exemple de fichier resolv.conf

```
nameserver 192.0.2.15
search example.com
```

## Connexion à une machine distante

### La commande SSH

La commande SSH permet de se connecter à distance à une machine et un port spécifiés

```
$ ssh [-p port] [usager@]machine
```

### Exemples:

- Les connexions au port 2223 de la machine locale seront renvoyées au port 23 de la machine 1, en passant pas un tunnel SSH entre la machine locale et la machine 2. À moins d'utiliser l'option `-g`, seule la machine locale peut utiliser ce renvoi.

```
$ ssh -L 223:machine1:23 machine2
```

- Les connexions au port 23 de la machine 2 seront renvoyées au port 23 de la machine, en passant par un tunnel SSH entre la machine 2 et la machine locale.

```
$ ssh -R 2223:machine1:23 machine2
```

- Redirige l'affichage graphique si autorisé avec un `X11Forwarding yes` dans `/etc/ssh/sshd_config` pour un serveur OpenSSH.

```
$ ssh -X [usager@]machine
```

## La commande scp

La commande `scp` permet d'effectuer une copie de fichier à distance sécurisée (même principe que la commande `cp`).

```
scp [[usager1]@machine1:]fichier [[usager2]@machine2:]chemin
```

Transférer récursivement un répertoire de manière sécurisée (utiliser l'option `-P` pour préciser un port particulier)

```
scp -R [[usager1]@machine1:]chemin [[usager2]@machine2:]chemin
```

## La commande sftp

La commande `sftp` permet le transfert defichier de manière sécurisée en mode interactif

```
sftp machine
```

## Archivage, signature, UUencode

### La commande gunzip

La commande `gunzip` décompacte une archive au format `.gz` ou `tgz`:

```
$ gunzip fichier.gz
```

### La commande unzip

La commande `unzip` décompacte une archive au format `.zip`:

```
$ unzip fichier.zip
```

## La commande uncompress

La commande **uncompress** décompacte une archive au format **.Z**:

```
$ uncompress fichier.Z
```

## La commande zip

La commande **zip** compacte au format **.zip**:

```
$ zip archive.zip fichiers
```

## La commande gzip

La commande **gzip** compacte une archive au format **.gz**:

```
$ gzip fichier
```

## La commande compress

La commande **compress** compacte une archive au format **.Z**:

```
$ compress fichier
```

## La commande bunzip

La commande **bunzip** permet d'extraire des archives au format **.bz2**

## La commande tar

La commande **tar** permet de manipuler des archives en respectant les droits la structure des répertoires archiver.

### Exemples:

- Créer une archive **.tar** d'un répertoire spécifié

```
$ tar -cf archive.tar repertoire
```

- Examine le contenu d'une archive spécifiée

```
$ tar -tf archive.tar
```

- Extrait le contenu d'une archive

```
$ tar -xf archive.tar
```

- Extrait l'archive dans un répertoire donné

```
$ tar -xf archive.tar -C repertoire
```

- Extrait une archive en respectant au mieux les droits d'accès aux fichiers contenus dans l'archive

```
$ tar -xpf archive.tar
```

- Décompacte et extrait une archive au format `.tgz` ou `tar.gz`

```
$ tar -xzf archive.tar.gz
```

- Décompacte et extrait une archive au format `.tgz` ou `tar.gz` dans le cas général

```
$ gunzip archive.tar.gz | tar -xf -
```

- Décompacte et extrait une archive au format `.tar.bz2` dans le cas général

```
$ tar -xjf archive.tar.bz2
```

## La commande sha1

La commande `sha1` affiche la signature SHA1 d'un fichier

```
$ sha1 fichier
```

**Remarque:** sous **linux** la commande s'appelle `sha1sum`.

**Exemple:** Comparaison des signatures SHA1 de deux fichiers

```
$ sha1sum fich1 fich2 | uniq | wc -l
```

## La commande md5

La commande `md5` affiche la signature MD5 d'un fichier.

```
$ md5 fichier
```

**Remarque:** sous **linux** la commande s'appelle `md5sum`.

## La commande uuencode

La commande `uuencode` encode un fichier au format UU

```
$ uuencode f > f.uu
```

## La commande uudecode

La commande `uudecode` décode un fichier au format UU

```
$ uudecode f.uu > f
```



## Gestion des processus

Commandes	Commentaires
ps	Affiche la liste des processus de l'utilisateur attachés à un terminal
ps -ax	Affiche tous les processus ( <b>BSD, Linux</b> )
ps -aux	Affiche tous les processus en indiquant les usagers ( <b>BSD, Linux</b> )
ps -ef	Affiche tous les processus ( <b>System V</b> )
top	Affiche les processus les plus gourmands ( <b>BSD, Linux</b> ). Pour réaliser l'affichage par ordre décroissant de mémoire utilisée option M.
systat	Affiche les processus les plus gourmands ( <b>BSD, MacOS X</b> )
nohup programme	Un lance un programme qui ne meure pas en fin de session
kill [-signal] pid	Tue un processus en donnant son PID
pkill [- signal] nom	Tue un processus en donnant son nom

### Signaux les plus souvent utilisés

Signaux	Commentaires
-1 (HUP)	Recharge le fichier de configuration
-2 (INT)	Interrompt le processus (résultat d'un <b>contrôle-C</b> )
-3 (QUIT)	Quitte le processus
-9 (KILL)	Tue le processus immédiatement (à éviter, essayer -15 avant)
-15 (TERM)	Tue le processus proprement (par défaut si pas de signal indiqué)
-18 (STOP)	Gèle le processus
-20 (CONT)	Reprend l'exécution d'un processus gelé

## Réaliser des tâches de maintenance périodiques

Commandes	Commentaires
crontab -l	Liste des tâches périodiques de l'utilisateur
crontab -e	Édite les tâches périodiques de l'utilisateur avec l'éditeur Vi
crontab -e -u usager	Édite les tâches périodiques d'un usager particulier (root uniquement)

### Format des commandes crontab

Une commande est composée de 5 champs temporels, d'un nom d'utilisateur s'il s'agit de la table générale du système et d'une commande. Les 5 champs temporels représentent les minutes (0-59), les heures (0-23), les jours du mois (1-31), les mois (1-12) et les jours de la semaine (0-7; le dimanche est représenté par 0 ou par 7):

- X,Y... Liste de valeur
- X-Y Intervalle de valeurs
- \* toutes les valeurs
- /n Période ou pas

## Exemples:

- Tous les jours: `0 5 * * *`
- Toutes les 5 minutes: `*/5 * * * *`
- Un mardi sur deux à 10h22: `10 22 * * 2/2`
- Du lundi au vendredi à 21h52: `21 52 * * 1-5`
- Le 10 et le 20 de chaque mois à 18h48: `18 36 10,20 * *`

## Gestion des disques et des partitions

Commandes	Commentaire
<code>mount</code>	Affiche la liste des systèmes de fichiers montés
<code>mount device rep</code>	Monte un système de fichiers (depuis un périphérique <code>device</code> ) sur un répertoire. <b>NB:</b> <code>device</code> est souvent de la forme <code>/dev/hd*</code> ou <code>/dev/sd*</code> .
<code>mount -t msdos device rep</code>	Montage d'une partition MS-DOS
<code>df</code>	Affiche l'espace occupé/libre sur les systèmes de fichiers montés: — <code>-k</code> Exprime les tailles en Ko — <code>-h</code> Exprime les tailles dans l'unité la plus significative ( <b>Linux, BSD</b> )
<code>fdisk device</code>	Modification du partitionnement
<code>pdisk device</code>	Modification du partitionnement ( <b>MacOs X, BSD</b> )
<code>disklabel device</code>	Modification du partitionnement ( <b>BSD</b> )
<code>newfs device</code>	Formatage d'une partition
<code>mkfs device</code>	Formatage d'une partition
<code>/etc/fstab</code>	Fichier contenant les correspondances entre devices et points de montage et indication des options de montage. <b>NB:</b> Le fichier <code>fstab</code> est également utilisé pour le montage automatique des partitions au démarrage.

## Développement

### Les commandes de bases

Commandes	Commentaires
<code>cc -c fichier.c</code>	Compilation de <code>fichier.c</code> en <code>fichier.o</code>
<code>cc -o prog f1.o f2.o</code>	Liaison de <code>f1.o</code> et <code>f2.o</code> pour générer le programme <code>prog</code>
<code>ar</code>	Manipulation d'archive <code>.a</code>
<code>ranlib</code>	Indexation des archive <code>.a</code>
<code>nm</code>	Liste les symboles d'un binaire
<code>objdump</code>	Affichage, désassemblage d'un binaire
<code>patch -pX fichier.patch</code>	< Applique le patch <code>fichier.patch</code> en supprimant <code>X</code> niveau d'arborescences dans les chemins

### Les principales options de GCC pour la compilation

Options	Commentaires
<code>-Wall</code>	Affiche tous les avertissements

<code>-Werror</code>	Prend les avertissements pour des erreurs
<code>ansi</code>	Exige un style ANSI
<code>-I/usr/pkg/include</code>	Recherche des entêtes dans <code>/usr/pkg/include</code>
<code>-O, -O2, -O3, -O6</code>	Optimisation pour la vitesse
<code>-Os</code>	Optimisation pour la taille de l'exécutable

## Les principales options de GCC pour la liaison

Options	Commentaires
<code>-static</code>	Liaison statique (utilise les librairies <code>.a</code> )
<code>-dynamic</code>	Liaison dynamique (utilise les librairies <code>.so</code> ou <code>.dylib</code> sous <b>MacOs X</b> ).
<code>-lbiblio</code>	Lie avec la libraririe <code>biblio.a</code> ou <code>biblio.so</code>
<code>-Lchemin</code>	Recherche les librairies dans <code>chemin</code>
<code>-Rchemin</code>	Indem, mais effectue la recherche au moment de l'exécution

**Remarque:** CC et un alias de GCC sous **BSD**, **Linux** et **MacOs X**.

## Utilisateurs

---

Commandes	Commentaires
<code>whoami</code>	Affiche son propre login
<code>who</code>	Affiche la liste des usagers
<code>finger usager</code>	Affiche les informations concernant un utilisateur
<code>talk usager[@machine]</code>	Engage une conversation avec un usager
<code>write usager</code>	Affiche un message sur le terminal d'un usager
<code>wall</code>	Affiche un message à tous les usagers
<code>msg [y n]</code>	Autorise (y) ou pas (n) l'affichage des messages
<code>tty</code>	Affiche le nom de son terminal

## Contrôle de version avec CVS et SVN

---

### Les commandes de bases de CVS

Commandes	Commentaires
<code>cvs -d dépôt co -P module</code>	Récupération d'un module depuis le dépôt
<code>cvs -d dépôt update -P -d</code>	Mise à jour des sources
<code>cvs -d dépôt diff</code>	Liste les differences au dépôt
<code>cvs -d dépôt diff   patch -R</code>	Annule les differences au dépôt
<code>cvs -d dépôt commit</code>	Valide les modification vers le dépôt

**Remarque:** L'option `-d` des commandes CVS peut être supprimée si la variable d'environnement `CVS_ROOT` est correctement initialisée.

### Les chemins de dépôt

---

Chemins	Commentaires
/chemin	Dépôt local
:pserver:usager@serveur:/chemin	Accès via le protocole pserver au dépôt sur le serveur <i>serveur</i> en utilisant le nom <i>usager</i>
usager@serveur:/chemin	Accès via le protocole rsh ou ssh en fonction de la valeur de la variable d'environnement CVS_RSH

## Les commandes de bases de SVN

Commandes	Commentaires
svn co <i>dépôt/dossier copie_locale</i>	Récupération d'un dossier depuis le dépôt
svn update <i>copie_locale</i>	Mise à jour des sources
svv diff <i>dépôt/dossier copie_locale</i>	Liste les differences au dépôt
svn <i>distant local</i>   patch -R	Annule les differences au dépôt
svn -commit <i>copie_locale</i>	Valide les modification vers le dépôt

### Exemple:

```
svn co https://www.ens.math-info.univ-paris5.fr/stage-unix/MementoUnix/ / home/mu/MementoUnix/
```