

# ADNAVIGO IRS / 3W.LOCOCHE.NET

## Commandes Unix / Linux

[cd](#) / [ls](#) / [cp](#) / [rm - rmdir](#) / [mkdir](#) / [mv](#) / [find](#) / [Autres commandes](#)

### Commande d'administration

- [Commandes Administration](#)
- [Messages Système](#)
- [La gestion des processus](#)
- [Éditer un fichier avec VI](#)
- [Recherche, analyse, modification](#)

### Arborescence UNIX classique

- [Organisation des répertoires](#)
- [Les Droits sous Linux](#)
- [Qu'est-ce qu'un shell ?](#)

### Commandes de Base Linux

- [Démarrage, redémarrage & arrêt](#)
- [Monter, démonter un système de fichiers](#)
- [Montage, démontage d'une partition Dos](#)
- [Montage du lecteur de CD-ROM](#)
- [Montage du lecteur de disquettes](#)
- [Copie d'une disquette](#)
- [Archivage](#)

---

### Se déplacer dans les répertoires (Change Directory)

[cd..](#) : Remonte d'un niveau  
[cd/](#) : Retourne à la racine  
[cd-](#) : Retourne au répertoire précédent

---

### Lister les fichiers d'un répertoire: (List Sorted)

[ls -l](#) : Permet de lister les attributs des fichiers  
[ls -d](#) : Affiche uniquement les répertoires  
[ls -a](#) : Liste tous les fichiers du répertoire y compris les fichiers cachés.  
[ls -m](#) : Affiche les fichiers en les séparant par une virgule.  
[ls -t](#) : Affiche les fichiers par date.  
[ls -lu](#) : Affiche les fichiers par date du dernier accès et indique la date.  
[ls -F](#) : Affiche les fichiers par type  
[ls -S](#) : Affiche les fichiers triés par ordre de taille décroissante.  
[ls -X](#) : Affiche les fichiers par type d'extension  
[ls -r](#) : Affiche les fichiers en ordre alphabétique inverse  
[ls -aLR /](#) : Affiche tous les fichiers d'un système  
[ls -aLR |grep doc](#) : Affiche tous les fichiers contenant doc

---

### Copier un fichier ou un répertoire: (copy)

[cp](#) : Demande s'il peut écraser le nom de fichier : répondre par Oui(y) ou Non (n)  
[cp -i](#) : Avertit de l'existence d'un fichier du même nom et demande s'il peut ou non le remplacer.  
[cp -l](#) : Permet de faire un lien en "dur" entre le fichier source et sa copie

**cp -s** : Permet de faire un lien "symbolique" entre le fichier source et sa copie  
**cp -p** : Permet lors de la copie de préserver toutes les informations concernant le fichier.  
**cp -r** : Permet de copier de manière récursive l'ensemble d'un répertoire et de ses sous répertoires  
**cp -b** : Permet comme l'option -i de s'assurer que la copie n'écrase pas un fichier existant : le fichier écrasé est sauvegardé, seul le nom du fichier est modifié et **cp** ajoute un tilde(~) à la fin du nom de fichier

---

Supprimer des fichiers et répertoires: (remove & remove directory)

**rm -d** : Permet de supprimer un répertoire qu'il soit plein ou non  
**rm -r** : Permet de supprimer un répertoire et ses sous répertoires  
**rm -f** : Permet de supprimer les fichiers protégés en écriture et répertoires sans confirmation  
**rmdir** : Supprime un répertoire  
**rmdir -p rep1/rep2/rep** : Supprime le répertoire et ses sous répertoires associés

---

Créer des répertoires: (make directory)

**mkdir** : Crée un répertoire  
**mkdir -p rep1/rep2/rep3** : Crée un répertoire et ses sous répertoires associés

---

Déplacer ou renommer un fichier: (move)

**mv -b** : Va effectuer une sauvegarde des fichiers avant de les déplacer  
**mv -i** : Demande pour chaque fichier et chaque répertoire s'il peut ou non le déplacer  
**mv -u** : Demande à "mv" de ne pas supprimer le fichier si la date de modification est la même ou plus récente que son remplaçant.

---

Recherche, analyse, modification de contenus

**find /** : Chercher à partir de la racine du fichier en cours  
**find -name** : Spécifier un nom de fichier  
**find -print** : Afficher le résultat à l'écran  
**find, locate et which** : Retrouver un fichier  
**diff** : Affichage des différences entre fichiers  
**cmp** : Comparaison binaire  
**cat et more** : Voir un fichier  
**sort** : Tri des lignes  
**df** : Emplacement du montage des systèmes de fichiers (partitions) et l'espace restant sur chacun d'eux  
**Sed** : Permet d'appliquer des commandes sur un fichier et d'afficher le résultat (sans modification du fichier)

**du** : Donne la place du disque utilisée par le répertoire courant  
**wc** : Comptage des mots, lignes ou caractères  
**file** : Donne le type de fichier  
**grep -c** : Trouver une expression rationnelle dans un fichier: (**grep**) donne le nombre de fois où l'expression rationnelle a été trouvée dans le fichier:  
**grep -n** : Pour rechercher des expressions rationnelles commençant par un tiret

---

#### Autres Commandes

**pwd** : Où suis-je ?  
**man** (et nom de la commande): Rechercher des informations sur une commande.  
**which** : Permet de connaître le chemin d'un exécutable  
**mdir** : Affiche la liste des fichiers d'une disquette MS-Dos  
**mkmode** : Crée un nom de fichier temporaire unique  
**emacs, joe** : Éditer un fichier  
**ln** : UNIX permet de créer plusieurs liens sur un même fichier avec la commande  
**who** : Liste les utilisateurs connectés sur la station  
**free** : Mémoire utilisée  
**clear** : Efface l'écran  
**touch** : Met à jour les dates d'accès des fichiers  
**logout** : Permet de fermer son compte utilisateur  
**logname** : Permet de savoir sous quel nom on est logué  
**tar** : Archivage de données : la commande  
**q** : quitte

---

#### Démarrage, Redémarrage & Arrêt

**startx** : demarrer Xwindows sous linux  
**reboot** : redémarre de la machine  
**shutdown -r** : arrête et redémarrer (rebooter).  
**shutdown -h** : arrêter proprement linux.  
**exit** : Déconnexion de l'utilisateur courant

---

#### Monter, démonter un système de fichiers

La commande **mount** est utilisée par linux dès son démarrage. Elle permet de monter un système de fichiers. Ce montage est parfois effectué automatiquement grâce au fichier de configuration **/etc/fstab**. Ce fichier contient tout ce que linux doit monter lors de son démarrage, mount n'est accessible que par root.

Montage, démontage d'une partition Dos (Ne doit pas être en cours d'utilisation)

1: `mount -t vfat /dev/hda1/dos/c`

2: `mount /dev/hda1/dos/c`

Pour convertir automatiquement les fins de lignes des fichiers ASCII ms-dos au format unix, utiliser l'option `conv`. `conv=binary/conv=auto -t vfat /dev/hda1 /dos/c`  
`mount -o conv=auto -t vfat /dev/hda1 /dos/c`

Démontage d'une partition Dos (Ne doit pas être en cours d'utilisation)

`umount /dos/`

Montage du lecteur de CD-ROM

1: `mount -t iso9660 /dev/hdb /cdrom`

2: `mount /dev/cdrom /cdrom`

On peut préciser l'option `-o` ou `-r` pour monter un périphérique en lecture seule, tel qu'un cdrom ou une disquette protégée en écriture par exemple.

Montage du lecteur de disquettes

`mount /dev/fd0 /floppy`

---

Copie d'une disquette

`cp /dev/fd0 disk.img` : du lecteur de disquette --> disk

`cp disk.img /dev/fd0` : d'un disk --> vers lecteur de disquette

`ln -s /répertoire/nom_source /nom_destination` : Création d'un lien symbolique

---

Archivage

`gzip` Compression et décompression des fichiers au format `gz`

`gzip groupe` : ce qui donne le fichier compressé `groupe.gz`

`gzip -d groupe.gz` : Décompression du fichier `groupe`

Options:

- `c` affiche le résultat de la compression sans modifier les fichiers originaux
- `d` décompresse le ou les fichiers
- `f` force l'écrasement sans demander confirmation
- `h` obtenir l'aide
- `l` affiche des informations sur le fichier
- `L` affiche la licence d'utilisation du programme
- `q` affiche les éventuels messages d'erreurs
- `r` pour agir sur les fichiers des sous-répertoires

- **S** modifie en .xxx l'extension des fichiers compressés
- **t** teste l'intégrité des fichiers compressés
- **V** affiche la version du programme
- [**x**] de 1 et 9 permet de choisir le taux de compression (1:faible, 9:maximum).

**tar**, Archiver plusieurs fichiers en un seul avec ou sans compression.

Syntaxe : **tar cvf nom\_de\_fichier.tar motif**

**tar cvf groupe.tar \*.c** : Création d'une archive qui porte le nom "groupe.tar" et qui contient tous les fichiers ayant l'extension **C**

**tar xvf groupe.tar \*.c**: Extraction de l'archive groupe.tar.

**tar zcvf groupe.tar \*.c**: Utilisation de l'option **z** pour compresser l'archive au moment de sa création.

**tar zxvf groupe.tar \*.c**: Décompression de groupe.tar pour Extraction des fichiers

Options:

**tar c groupe.tar \*.c**: pour archiver

**tar x groupe.tar**: pour dé-archiver

**tar cz groupe.tar.gz \*.c**: pour archiver et compresser

**tar xz groupe.tar.gz**: pour décompresser et extraire.

---

Commande d'administration système Linux

**at** : Commande permettant de lancer une autre commande ou un processus en différé.

**cron** : Outil logiciel disponible sous Unix permettant de lancer périodiquement les processus indiqués dans la crontab.

**crontab** : table contenant des commandes qui doivent être exécutées périodiquement. C'est le programme cron qui lance la commande. Les périodes peuvent être des heures (cron.hourly), des jours (cron.daily), des semaines (cron.weekly) ou des mois (cron.monthly).

**chmod** : Placer les droits d'utilisation des fichiers

**chown** : Désigner l'utilisateur et le groupe propriétaire des fichiers

**adduser** : Ajouter un utilisateur

**passwd** : Spécifier ou modifier un mot de passe

**chfn** : Décrire un utilisateur

**userdel** : Supprimer un utilisateur

**mount** : Utiliser votre CD-ROM, votre lecteur de disquette

**shutdown -h** : Arrêter le système

**mkbootdisk (device /dev/fd0 2.2.13-4.mdk)**: Créé une disquette de Boot

**lilo -u** : Désinstalle LILO

## Messages Système

`cat /proc/interrupts` : Affiche les IRQ utilisés  
`cat /proc/ioports` : Affiche les ports I / O utilisés  
`cat /proc/dma` : Affiche l'utilisation des canaux DMA  
`cat /proc/pci` : Affiche l'utilisation des ports PCI

---

## La gestion des processus

`top` : permet de suivre les ressources que le processus utilise  
`ps` : permet de connaître les processus actifs à un moment donné  
`pstree` : permet d'afficher les processus sous forme d'arborescence et donc de voir leurs interdépendances  
`kill` : Permet de tuer un processus en court : `syntaxe kill [option] PID`.  
Pour tuer le processus, je peux d'abord faire un `ps -ax` pour connaître le numero du PID et ensuite si par exemple le PID est 3600, je peux tuer la connexion en faisant  
`:[root@localhost/root]# kill 3600`

---

## Éditer un fichier: (éditeur vi)

Passer du mode commande aux mode texte, taper du mode texte, enregistrer.

vi comprend deux modes : un mode "commande" et un mode "insertion", après le lancement de vi nous sommes en mode commande : appuyez sur la touche "Echap" puis sur "a" ("a", comme "append", permet d'ajouter du texte après le curseur). Vous voyez en bas de l'écran apparaître la ligne "INSERT".

**Sauvez un fichier** : Il faut sortir du mode texte en appuyant à nouveau sur la touche "Echap". La mention "INSERT" disparaît, nous sommes en mode commande. Tapez maintenant `:"w nomfichier"` et sur la touche retour chariot (afin d'écrire ("`write`") le fichier). Vous devez obtenir en bas de l'écran ceci : "nomfichier" [New File] 3 lines, 142 characters written

## Supprimer du texte et quitter vi

J'ai fait une faute d'orthographe! Nous allons supprimer le "p" qui est en trop dans "suppprime" : déplacez le curseur sur un des "p" en trop, passez en mode commande ("INSERT" ne doit pas apparaître à l'écran), appuyez sur "x", le "p" a disparu.

Sauver les modifications effectuées : Passez en mode commande et tapez `:"wq"` (write et quit). Vous êtes sorti de vi et votre fichier a été sauvegardé sous linux-test. Pour revenir à vi en ouvrant le fichier linux-test au démarrage tapez : `:[root@localhost/root]# vi linux-test`

Si vous souhaitez quitter sans enregistrer les dernières modifications, il vous faudra passer en mode commande et taper `:"q!"`.

D'autres commandes vi.

**A** : permet d'ajouter du texte à la fin de la ligne.

**i** : permet d'ajouter du texte avant le curseur.

**o** : permet d'ajouter une ligne en dessous du curseur.

**O** : permet d'ajouter une ligne au dessus du curseur.

**le retour chariot** : permet d'aller à la ligne suivante.

**dd** : permet de supprimer la ligne courante.

**X** : permet de supprimer le caractère avant le curseur.

**u** : permet d'annuler la dernière commande effectuée.

---

Organisation des répertoires - arborescence d'un système UNIX classique

**/** : Répertoire racine, tous les autres répertoires en dépendent.

**/bin** : contient les binaires fondamentaux à la gestion de Linux.

**/dev** : contient une multitude de fichiers dits spéciaux. L'un deux correspond à mon modem. Je dois indiquer ce fichier dans la configuration de mes outils de communication. De même **/dev/hda1** correspond à la première partition de mon disque dur IDE, si mon disque dur est un SCSI, son nom sera **/dev/sda1**. Un dernier exemple : **/dev/fd0** correspond à mon lecteur de disquettes.

**/etc** : contient tous les fichiers de configuration de linux. On y retrouve par exemple le fichier **/etc/passwd**, qui définit les mots de passe des utilisateurs.

**/sbin** : contient les binaires du système. On y trouve par exemple la commande **shutdown** qui permet d'arrêter l'ordinateur.

**/home** : répertoire qui contient les répertoires des utilisateurs du système. Le répertoire des utilisateurs est automatiquement créé avec la création d'un compte. Tous mes fichiers personnels sont dans **/home/(maltesse)**.

**/lost+found** : est le répertoire des fichiers perdus. Ces fameux fichiers qui, du fait d'erreur disque, se retrouvent sans chemin d'accès. Le binaire **fsck**, qui est lancé régulièrement au démarrage de linux, se charge de les détecter et de les stocker dans le répertoire **/lost+found**

**/tmp** : est un répertoire accessible par tous les utilisateurs du système, il permet de ne pas encombrer son répertoire personnel par des fichiers que l'on souhaite de toute manière détruire ou modifier.

**/var/spool** : répertoire des fichiers qui servent de file d'attente. Par exemple, les files d'attente de l'imprimante se trouvent sous ce répertoire. Les données à imprimer, envoyer, ... sont stockées dans ces files d'attentes jusqu'à ce qu'elles soient traitées.

**/usr** : contient tout ce qui concerne les binaires utiles à tous les utilisateurs et quelques commandes d'administration. On y trouve cependant d'autres choses: **/usr/bin** contient donc les binaires disponibles pour les utilisateurs et les scripts.

**/usr/X11R6** : contient tout ce qui concerne Xfree86 (les bibliothèques, les binaires, la documentation).

**/usr/include** : contient tous les "headers" nécessaires à la programmation dans les différents langages.

**/usr/lib** : contient toutes les bibliothèques nécessaires au fonctionnement des logiciels. (comme par exemple la bibliothèque C ou C++ ou tcl/tk).

**/usr/local** : on y met ce que l'on veut, mais surtout les fichiers d'usage local.

---

## Les Droits sous Linux

**Droit de lecture (r pour Read)** Pour un fichier: Autorise la lecture du contenu. Pour un répertoire: Autorise à lister les fichiers contenus dans ce répertoire, si le droit d'exécution sur ce répertoire est positionné également.

**Droit d'écriture (w pour Write)** Pour un fichier: Autorise à en modifier le contenu. Pour un répertoire: autorise à créer des fichiers et à en effacer, même si l'on n'est pas le propriétaire de ces fichiers.

**Droit d'exécution (x pour exécute)** Pour un fichier: Autorise l'exécution. Pour un répertoire: Autorise l'utilisateur à le traverser, c'est-à-dire de s'y rendre ou de se rendre dans l'un de ses répertoires fils.

champ 1	2	3	4	5	6	7
<b>-rwxr-xr-x</b>	1	Dupond	labo2	3528	04 août 02 10:10	linux
<b>drwxrwxr-</b>	3	Dupond	labo2	512	02 jan 2002	conso
<b>lrwxr-xr--</b>	2	Dupond	labo2	210	16 dec 01 10:30	awk --> gawk

Définition des champs du tableur ci-dessus:

1 - Le premier (-) indique un fichier ou (d) indique un répertoire ou (l) indique un lien (raccourci). En suite par groupe de trois nous avons d'abord sur la première ligne:

Les droits du propriétaire (rwx) lecture, écriture, exécution	<b>7</b>
Les droits groupe (r-x) lecture, exécution	<b>5</b>
Les droits de tout le monde (r-x) lecture, exécution	<b>5</b>

2 - Indique le nombre fichier, de répertoire ou de lien.

3 - Indique le nom du propriétaire du fichier, du répertoire ou du lien.

4 - Indique le nom du groupe aux quels appartiennent fichiers & répertoire.

5 - Indique la taille.

6 - Indique la date de dernière modification & l'heure de création dans le cadre d'un (ls -l).



7 - Indique le nom du fichier, répertoire ou lien

C'est la commande `chmod` qui permet de modifier les permissions qu'ont les utilisateurs sur les fichiers. Évidemment, seul le propriétaire du fichier a le pouvoir de modifier ces permissions (à part bien sûr le super utilisateur "`root`" qui peut faire absolument tout ce que bon lui semble).

Par exemple, décidons que n'importe qui pourra modifier le fichier linux :

```
[dupont@localhost/dupont]$ chmod a+w linux
```

"a" indique que tous les utilisateurs seront touchés par la modification des permissions

"+" signifie que c'est une permission supplémentaire que l'on donne. Pour en supprimer une il suffit de remplacer le signe "+" par "-".

"w" signifie que c'est la permission d'écriture que nous donnons.

Pour vérifier que tout a bien fonctionné, faites un "`ls -l linux`", nous obtenons `:-rw-rw-rw- 1 dupont dupont 04 août 02 10:10 linux:`

Si maintenant nous voulons supprimer ce droit d'écriture mais aussi le droit de lecture pour le groupe propriétaire et les autres utilisateurs nous utilisons la syntaxe suivante :

```
[dupont@localhost/dupont]$ chmod go-wr linux
```

"go" signifie que la commande affectera le groupe propriétaire et les autres utilisateurs.

"wr" signifie que la modification portera sur les droits d'écriture ou de lecture.

#### Chmod 644 - droit d'accès par défaut

Propriétaire	lecture	<b>X</b>	écriture	<b>X</b>	exécution	4+2+0	<b>6</b>
Groupe	lecture	<b>X</b>	écriture		exécution	4+0+0	<b>4</b>
Tous	lecture	<b>X</b>	écriture		exécution	4+0+0	<b>4</b>

#### Chmod 755 - droit d'accès le plus utilisé

Propriétaire	lecture	<b>X</b>	écriture	<b>X</b>	exécution	<b>X</b>	4+2+1	<b>7</b>
Groupe	lecture	<b>X</b>	écriture		exécution	<b>X</b>	4+0+1	<b>5</b>
Tous	lecture	<b>X</b>	écriture		exécution	<b>X</b>	4+0+1	<b>5</b>

#### Chmod 777 - droit d'accès total (root)

Propriétaire	lecture	<b>X</b>	écriture	<b>X</b>	exécution	<b>X</b>	4+2+1	<b>7</b>
Groupe	lecture	<b>X</b>	écriture	<b>X</b>	exécution	<b>X</b>	4+2+1	<b>7</b>
Tous	lecture	<b>X</b>	écriture	<b>X</b>	exécution	<b>X</b>	4+2+1	<b>7</b>

Qu'est-ce qu'un shell ?

Sous UNIX: C'est un interpréteur de commande (en mode texte) la partie du système d'exploitation utilisé comme interface avec l'utilisateur. Originellement utilisé sous Unix, s'est répandu un peu partout depuis. Il permet l'exécution de script Bash (suite de commandes d'un fichier exécutable).