

Mémo des commandes Unix

Dans une machine tournant sous Unix, les fichiers sont organisés selon une arborescence, basée dans un répertoire dit *racine*, dont le nom est simplement “/”. Les sous-répertoires successifs peuvent être atteints en utilisant des caractères slash : `/home/toto/mon_fichier` désigne le fichier `mon_fichier`, présent dans le répertoire `toto`, lui-même situé dans le répertoire `home` qui est lui-même situé à la racine. Le répertoire `/home/` regroupe les répertoires des différents utilisateurs. Ici `/home/toto/` est le répertoire personnel de l'utilisateur `toto`.

Dans un répertoire donné, la syntaxe “..” désigne le répertoire parent, et la syntaxe “.” désigne le répertoire courant. Par exemple, si l'on est dans le répertoire `/home/toto/`, la commande `ls .` va lister tous les fichiers présents dans le répertoire `/home/toto/`, et la commande `ls ..` va lister tous les fichiers présents dans le répertoire `/home/`. De même, si le répertoire courant est `/home/toto/dossier_1/` on écrira `cd ../../titi/dossier_2` pour se rendre dans le répertoire `/home/titi/dossier_2`.

Voici une liste (assez arbitraire) des commandes Unix les plus courantes.

Commande	Effet	Exemple
<code>man</code>	Affiche le manuel (l'aide) d'une commande	<code>man commande</code>
<code>exit</code>	Quitte la ligne de commande	<code>exit</code>
<code>pwd</code>	Affiche le nom du répertoire courant	<code>pwd</code>
<code>cd</code>	Change le répertoire courant	<code>cd ../un_repertoire</code> (<code>cd</code> tout seul permet de retourner dans son répertoire personnel)
<code>ls</code>	Liste les éléments d'un dossier	<code>ls home/mon_repertoire</code> ou simplement <code>ls</code>
<code>rm</code>	Supprime un fichier	<code>rm fichier</code>
<code>mkdir</code>	Crée un répertoire	<code>mkdir nom_du_repertoire</code>
<code>rmdir</code>	Efface un répertoire vide	<code>rmdir nom_du_repertoire</code>
<code>mv</code>	Déplace un fichier ou un répertoire	<code>mv ancien_nom nouveau_nom</code>
<code>cat</code>	Affiche le contenu d'un fichier	<code>cat fichier</code>
<code>head</code>	Affiche le début du contenu d'un fichier	<code>head fichier</code>
<code>grep</code>	Cherche une suite de caractères dans un fichier	<code>grep toto fichier</code> cherche “toto” dans <code>fichier</code>

<code>bg</code>	Fait tourner en arrière-plan un programme arrêté avec Ctrl+Z	<code>bg</code>
<code>fg</code>	Fait tourner au premier plan un programme arrêté avec Ctrl+Z	<code>fg</code>
<code>ps</code>	Affiche la liste des processus (= des programmes actifs) ainsi que leurs indentifiants (PID)	<code>ps</code>
<code>kill</code>	Tue un processus identifié par son PID	<code>kill 21703</code>
<code>pidof</code>	Obtenir le PID d'un processus	<code>pidof firefox</code>
<code>pkill</code>	Tue un processus identifié par son nom	<code>pkill firefox</code>

Il est possible de faire tourner un processus en arrière-plan en ajoutant le caractère `&` après la commande qui le lance. Par exemple, entrer `gedit &` va lancer le programme `gedit`, mais la ligne de commande sera encore disponible.

On peut arrêter le processus au premier plan avec Ctrl+Z. Ce processus pourra être relancé avec `fg` ou `bg`. Pour tuer le processus au premier plan, on utilise la combinaison Ctrl+C.