

Quelques commandes utiles mentionnées dans ce cours

- Rappel : « `man commande` » fournit une notice d'utilisation de la *commande*, à commencer par « `man man` »...
- `alias` : permet de donner des diminutifs à des commandes souvent utilisées. `alias nom="vraie commande"`
 - `bash` : le shell le plus courant, souvent appelé simplement `sh`.
 - `cat` est plus frustré que `less` : il recopie sans mise en page, sur sa sortie standard, le contenu du fichier qu'on lui donne en argument.
 - `cd` : change directory (change le répertoire de travail courant).
 - `chmod` : pour attribuer les droits d'accès à un fichier dont on est propriétaire (change mode).
 - `echo` : écrit les chaînes de caractères qu'on lui donne en argument (utile pour connaître la valeur d'une variable par exemple).
 - `evince`, `xpdf`, `epdfview...` : outils de visualisation en impression de fichier PDF (versions open-source de « readers » bien connus qui présentent souvent des trous de sécurité tant on ne sait pas ce qu'ils font)
 - `file` : détermine le type d'un fichier en explorant son contenu réel.
 - `firefox` : navigateur internet très configurable
 - `gedit` : éditeur de fichiers textes d'utilisation assez simple, bien adapté pour écrire des shell-scripts.
 - `gimp` : programme très puissant de manipulation d'images (similaire photoshop).
 - `grep` : affiche à l'écran toutes les lignes de son entrée standard ou d'un fichier contenant à un motif (voir section du cours à ce sujet) donné en argument.
 - `k3b` ou autres graveurs de CDROM, DVD, BluRay : utiles aussi pour sauvegarder les fichiers `tar` de la commande précédente...
 - `kompozer` : éditeur convivial de pages HTML
 - `less` : montre un fichier texte page par page dans le terminal, la touche *espace* permettant de passer à la page suivante.
 - `lpr` : imprime le fichier qu'on lui donne en argument sur l'imprimante désignée par la variable `$PRINTER`.
`lpq` : montre la liste d'attente de l'imprimante et/ou indique les problèmes d'impression
`cancel` ou selon les systèmes `lprm` : supprime une demande d'impression de la file d'attente.
 - `ls` : liste le contenu d'un répertoire ; cette commande admet de nombreuses options, dont `-a` (all) pour afficher aussi les « fichiers cachés » c'est-à-dire ceux qui commencent par un point, et aussi `-l` (format long) pour afficher les informations principales à propos des fichiers (droits, propriétaire, taille, etc).
 - `mimetype` : indique le contenu probable d'un fichier en utilisant son suffixe éventuel
 - `mkdir` : crée un répertoire
 - `mv` : « move » un fichier (mais en fait, se contente de le renommer).
 - `openOffice` ou `freeOffice` : versions open-source des suites bureautiques du commerce.
 - `ps` : fournit la liste des processus lancés à partir du terminal par l'utilisateur. Les options « `-x` » et « `-aux` » montrent plus de processus (faire `man ps`).
 - `pwd` : print working directory
 - `rm` : supprime (définitivement, il n'y a pas de « corbeille ») un fichier plat ou un lien symbolique. Avec l'option `-r` (`r` comme récursif), « `rm -r repertoire` » supprime le *répertoire* et tout son contenu.
 - `sed` : stream editor, prend en argument des commandes de remplacement de texte (faire `man sed` en portant surtout attention à l'option `-e` et à la commande de remplacement de la forme « `s/vieux/nouveau/g` »).
 - `tar` : permet de créer un fichier d'archive de toute une arborescence de fichiers. Utile pour les sauvegardes ou pour les échanges par mail de données structurées.
 - `thunderbird` : gestionnaire de mails.
 - `top` est plus sophistiqué que `ps` : il fournit en temps réel la liste des processus les plus gourmands en puissance de calcul (ordre décroissant) de l'ordinateur. On en sort avec la touche « `q` ».
 - `touch` : positionne les dates de dernière modification et de dernière utilisation d'un fichier à l'instant présent.
 - `which` : prend en argument un nom de commande et explore la variable `$PATH` pour fournir l'adresse de son fichier exécutable.
 - `whoami` : affiche votre nom d'utilisateur sur la machine.
 - `xterm` ou `kterm` ou autre « fenêtre de terminal » : fait apparaître dans une fenêtre graphique un espace textuel qui permet d'utiliser le shell.