

Introduction à GNU/Linux et à la ligne de commande

Simon Liénardy

30 septembre 2019



Système d'exploitation (1/2)

- Un *système d'exploitation* (operating system, OS) est un programme permettant une utilisation simple et efficace des ressources d'un ordinateur¹.



Principalement :

- Le(s) processeur(s)
- La mémoire
- Les périphériques

Systeme d'exploitation (2/2)

- Exemples : GNU/Linux, Windows, Mac OS,...
- Et dérivés : iOS, Android,...
- GNU/Linux = noyau (ou *kernel*) Linux + logiciels GNU.
 - Noyau ?

Partie la plus centrale du système
 - Logiciels **GNU**

=> **g**libc, **g**cc sans lesquels faire tourner un programme serait très difficile.

Distribution GNU/Linux

- Et Ubuntu ? Linux Mint ? Fedora ? CentOS ? Debian ?
=> Ce sont des *distributions GNU/Linux*

Distribution = GNU/Linux + programmes

Différence entre les distributions ?

- Programmes : (non-)libre ? Spécialisé / Généraliste ?
- Politique de mise à jour (stable vs m.à j. fréquentes)
- Bureau / serveur / autre (ex. : smart TV)
- Mécanisme de dépôts de paquets
- Etc.

Ligne de commande

- La ligne de commande permet d'interagir avec les programmes et le système d'exploitation
- Le programme qui interprète les commande s'appelle un shell
- Logiciel le plus courant : bash
- Interpréter ?
 - `nom@machine:~$` attend des instructions
 - vérifie la syntaxe
 - rend ces instructions compréhensibles à l'ordinateur

Ligne de commande : exemple

- Exemple :

```
pmli1814@candi17 :~$ python fichier.py
```

↓
Votre login ou nom
d'utilisateur

↓
Le nom de
la machine

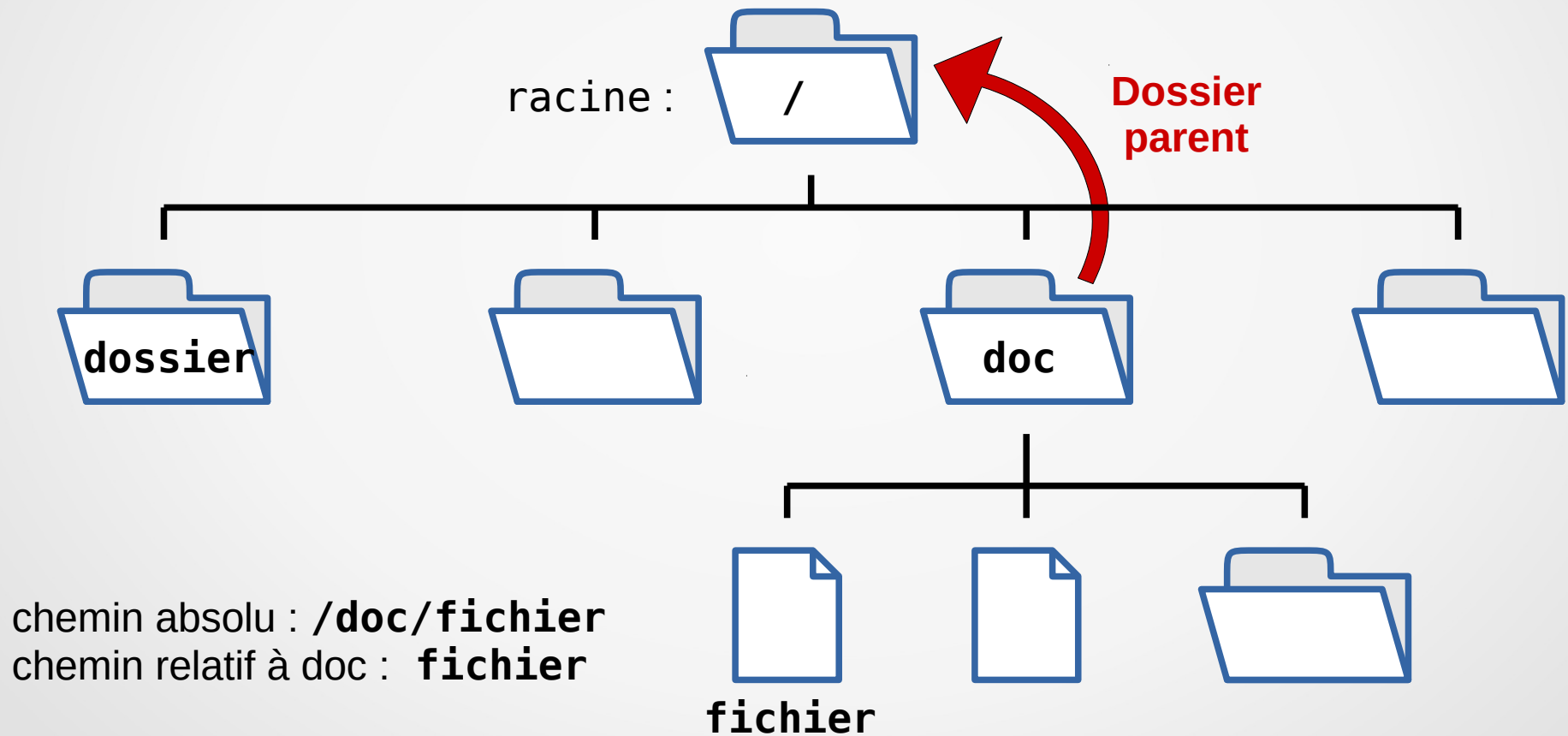
↓
Localisation dans
l'arborescence des
fichiers

↓
Le nom du
programme

↓
Paramètre(s) du
programme

Systeme de fichier

- Structure arborescente



Premières commandes

- `ls [Chemin]` liste le contenu d'un répertoire
 - `ls -a` liste aussi les fichiers cachés (`.nom_caché`)
 - `ls -l` affiche des informations sur les fichiers :

drwxr-xr-x 3 user group 4.0K mai 21 17:39 dossier

Nombre de
liens physiques

Taille

Date de dernière
modification

Nom du
dossier

Permissions : - Read, Write, Execute pour utilisateur, groupe et autres utilisateurs
- d = Directory (c'est un répertoire), execute = ouvrir le répertoire

- `ls -la` combine les 2 opérations

Premières commandes

- `cat nom_fichier` pour lire un fichier dans le terminal
 - `cat *.c` argument = tous les fichiers qui se terminent par « .C »
- `cd [CHEMIN]` se déplacer dans le répertoire CHEMIN
 - `~` est le répertoire de l'utilisateur (alt gr + '=' ou alt + 'n')
 - `.` est le répertoire courant (`cd .` ne fait donc rien)
 - `..` est le répertoire parent dans la hiérarchie
 - exemple : `cd /usr/include`
- `pwd` affiche le répertoire courant

ls / : Filesystem Hierarchy Standard

bin/	Commandes essentielles
boot/	Fichiers de chargement de l'OS
dev/	Devices (matériel)
etc/	(<u>E</u> ditable <u>T</u> ext <u>C</u> onfiguration) fichiers de configuration
home/	Répertoire des utilisateurs
lib/	Librairies essentielles
lib64/	Idem compilées en instructions 64 bits
media/	Point de montage des medias amovibles (ex.: clés usb, shared files in VBox)

ls / : Filesystem Hierarchy Standard

mnt/	Point de montage (<u>m</u> ount) temporaire
opt/	Logiciels optionnels installés pour tous les utilisateurs
proc/	Système de fichier virtuel contenant des info sur les processus, le noyau,...
root/	Répertoire personnel du super-utilisateur
sbin/	<u>S</u> ystem <u>b</u> inaries = commandes système essentielles
tmp/	Fichiers <u>t</u> em <u>p</u> oraires (vidé par défaut à chaque démarrage du système)
usr/	<u>U</u> nix <u>S</u> ystem <u>R</u> essources
var/	Fichiers <u>v</u> ariables (cache, logs, ...)

ls /usr/

bin/	Binaires des programmes non-essentiels à un système minimal (ex. : gcc)
include/	En-tête des bibliothèques partagées (ex. : stdio.h)
lib/	Bibliothèques partagées (ex. : libm.so)
local/	Sous-hiérarchie locale à l'ordinateur
man/	Contenait les manpages (maintenant dans /usr/share/man)
sbin/	Binaires systèmes non-essentiels
share/	Données indépendantes de l'architecture (partageable) (ex. : manpages)
src/	Sources des programmes

Avant de poursuivre : obtenir de l'aide

- `man NOM`
- fonction de la **bibliothèque standard** :
 - `man 3 printf`
- man pages situées dans `/usr/share/man`
- Possible de les installer en français.
- La version anglaise est la version de référence
- taper `/TEXTE (+ enter)` pour rechercher TEXTE, naviguer entre les occurrences avec `'N'` pour aller à l'occurrence suivante, `MAJ + 'N'` pour la précédente

Obtenir de l'aide 2 : *Google is your friend*

- N'hésitez pas à chercher de l'aide sur internet
- En anglais pour parfaire votre connaissance de la langue (vous trouverez aussi **plus de résultats**)
- Google ou **autres** (Si c'est gratuit, c'est vous le produit !
→ Qwant, DuckDuckGo, Ecosia,...)
- Rechercher intelligemment avec un (méta-)moteur :
 - recherche avancée
 - connecteurs logiques (OR, AND, (...))
 - "expression exacte"
 - string -lingerie
 - string langage C

Gestion des fichiers et dossiers

- `touch nom` crée un fichier nommé `nom`
- `mkdir nom` = make directory = créer répertoire
 - `mkdir "nom avec espace"`
 - `mkdir -p creer/aussi/parents`
- `mv f1 [f2 ...] rép` déplacement des fichiers `fi` dans le répertoire `rép`
 - `mv` permet aussi de renommer un fichier :
`mv nom_fichier nouveau_nom`

Gestion des fichiers : suite

- `cp source destination` copie de fichiers
 - `cp -R rep_source dest` copie de répertoires
- `rm fichier` Supprimer un fichier (remove)
- `rm -r repertoire` Supprimer un répertoire et son contenu (r pour récursif)

Gestion des fichiers : suite

- `sudo rm -r /*`
 - Effet de cette instruction ?
 - Tout effacer
 - Qu'est-ce qui l'en empêche ?
 - Rien (si sudo est mal configuré) !
 - Morale de l'histoire :



Un grand pouvoir implique de grandes responsabilités !

Shell et caractères spéciaux (1/2)

Caractère spécial	Interprétation par le shell
* (joker)	Tous les fichiers d'un répertoire
' ' (l'espace)	Sépare 2 arguments
. (début d'un chemin)	Répertoire courant
..	Répertoire parent
/	Sépare les répertoires dans les chemins
# (<i>croisillon</i>)	Introduit un commentaire
\$var	Demande le contenu de la variable var
\$?	Code retour de la commande (cf. <code>return 0;</code>)
> (et variantes)	Redirection (voir infra)
\ (barre oblique inversée)	Annule l'interprétation du caractère spécial suivant

Shell et caractères spéciaux (2/2)

- Et autres : () [] { } ? " ' , & ! |

- **ATTENTION !** : *ne pas les utiliser dans des noms de fichiers. Sinon, les faire précéder d'un backslash (\) lors de l'appel ; ou « quoter » le nom : "comme ceci".*

Autant n'utiliser que des chiffres, des lettres, le tiret et l'underscore (_). C'est le cas pour les **Challenges**

Modifier des permissions : chmod

- `chmod mode fichier`
 - `mode = {u,g,o,a}{+,-, =}{r, w, x, rw, rx, rwx,...}`
- Exemple :
 - `chmod o+x fichier` ajoute le droit d'exécution aux utilisateurs n'appartenant pas au groupe (les autres)
 - `u` = propriétaire, `g` = groupe, `o` = les autres, `a` = tout le monde
 - `+` : ajout de permission, `-` : retrait de permission,
`=` : autorise exclusivement ce qui est spécifié
 - `chmod a=rwx *` : mauvaise politique de sécurité

Lire et parfois même éditer (!) un fichier

- `cat fichier` affiche le fichier sur le terminal
- `less fichier` permet de naviguer dans le fichier
- `nano fichier` éditeur simple dans le terminal
- `emacs / vim` pour les masochistes ou ceux qui n'ont pas accès à une GUI (connexion à un serveur par ex.)
- `gedit pgrm.c &` ouvre `pgrm.c` dans `gedit` et ne bloque pas le terminal (symbole `'&'` en fin de ligne)

À préférer en 2019 !

Préférences de gedit

Affichage

Éditeur

Police et couleurs

Greffons

Tabulations

Largeur des tabulations :

3

–

+



Insérer des espaces au lieu de tabulations



Activer l'indentation automatique

Enregistrement des fichiers



Créer une copie de sauvegarde des fichiers avant l'enregistrement



Enregistrer automatiquement toutes les

10

–

+

minutes

Aide

Fermer

Rechercher un fichier

- `locate PATTERN`
 - `PATTERN` peut être une suite de caractères ou une expression plus complexe
 - nécessite de créer une base de donnée avec `updatedb` (doit être s.u. pour indexer tous les rép.)
- `find /chemin -name PATTERN`
 - plusieurs chemins peuvent être spécifiés, `find` cherche dans l'arborescence à partir de chaque chemin
 - Il existe d'autres expressions de recherche que « `-name PATTERN` » (recherche en fonction de dates, propriétaires, nbre liens, ...)

Recherche dans les fichiers textes

- `grep TEXTE fichiers`
 - recherche la chaîne de caractère `TEXTE` dans les fichiers fournis en argument
- `grep -r TEXTE`
 - recherche de manière récursive dans tous les fichiers appartenant au répertoire (cherche dans les répertoire contenu dans le répertoire, ...)
- Outil très puissant si utilisé avec l'ensemble de ses options.

Autres opérations sur les fichiers

- Remplacer du texte (entre-autre) avec `sed`
- Comparer deux fichiers avec `diff`
- Télécharger un fichier :
 - `wget url`
 - plus d'options qu'un navigateur web et plus robuste aux connections lentes ou instables
 - permet d'aspirer des sites web (= conserver la hiérarchie des pages du sites et conserver la consistance des liens entre les pages téléchargées).

Compression et archivage

- Une archive est un fichier qui contient plusieurs fichiers et répertoires
 - Très souvent compressée pour réduire la place occupée
- `tar -czf nom.tar.gz fichier1 dossier2`
 - `c` : créer
 - `z` : utiliser gzip pour la compression
 - `f` : spécifier le nom du résultat (`nom.tar.gz`)
- `tar -xzf nom.tar.gz`
 - Opération inverse

Compression et archivage

- Pour le 1^{er} Challenge :

```
zip challenge1.zip challenge1.txt
```

- Pour décompresser :

```
unzip archive.zip
```

Pipe et redirections

- Les programmes peuvent interagir avec 3 flux standards :
 - `stdin` l'entrée standard (par ex. : lue par `scanf`)
 - `stdout` la sortie standard (ex. : sortie de `printf`)
 - `stderr` la sortie d'erreur standard
- On veut pouvoir rediriger ces flux vers un fichier

Pipe et redirections

- `pgm > file` redirige `stdout` vers le fichier `file`
- `pgm >> file` redirige `stdout` vers le fichier `file` (ajoute à la fin de `file`)
- `pgm 2> file` redirige `stderr` vers le fichier `file`
- `pgm &> file` redirige `stderr` et `stdout` vers `file`
- `pgm 2>&1` redirige `stderr` vers `stdout`
- `prgm < file` lit `file` plutôt que `stdin`

Pipe

- Faire en sorte que la sortie standard d'un programme soit l'entrée standard d'un programme suivant :

`prgm1 | pgrm2`

- `|` est appelé « pipe »
- `Alt gr + &` sur les claviers AZERTY belges
- `Alt + Maj + L` sur les claviers Mac
- ex. : `ls /usr/include | grep std`

Script

- Pipes et redirections permettent de combiner des résultats de commande
- On peut aller plus loin, bash permet :
 - d'écrire des instructions conditionnelles
 - d'écrire des boucles
 - de manipuler des variables, des tableaux, ...
- C'est un véritable langage qui permet d'écrire des programmes (scripts)
 - <https://openclassrooms.com/courses/reprenez-le-contrôle-a-l'aide-de-linux/introduction-aux-scripts-shell>
 - <http://www.gnu.org/software/bash/manual/bashref.html>

Installer de nouveaux programmes

- Commandes dépendent du gestionnaire de paquet : apt / dnf
- Chercher si pgrm est disponible
 - `apt(-cache) search pgrm`
 - `dnf search pgrm`
- Installer un programme (s.u. nécessaire)
 - `apt(-get) install pgrm`
 - `dnf install pgrm`
- Désinstaller un programme (idem)
 - `apt(-get) remove pgrm`
 - `dnf remove pgrm`

Installer ?

- Appeler gcc produit un exécutable dans le répertoire courant.
→ nécessaire de spécifier le chemin pour l'utiliser. ex. : `./a.out`
- Le gestionnaire de paquets s'occupe des dépendances, de positionner l'exécutable au bon endroit dans l'arborescence de fichier, etc.
- Variable de chemin : PATH (afficher grâce à `echo $PATH`)

Avant d'appeler une commande, le shell cherche si elle n'est pas dans le chemin.

Ex. : `/usr/bin/ls` abrégé en `ls` car `/usr/bin` est dans PATH

Travailler à distance avec SSH

- Travailler à distance avec un client ssh (**Secure Shell**) :
`ssh -X pml1234@candiYY.montefiore.ulg.ac.be`
 - -X permet de lancer des pgrm en mode graphique
 - Remplacez 1234 par votre pml
 - YY est le numéro d'une machine
- Transférer des fichier avec scp (exécutée sur **votre** ordi):

- Télécharger :

`scp pmlXXXX@candiYY.montefiore.ulg.ac.be:./filepath/src dest`

- Téléverser :

`scp src pmlXXXX@candiYY.montefiore.ulg.ac.be:./path/dest`

Attention au ':' !

Quelques tuyaux

- Arrêter un programme
 - Ctrl + C
 - Ctrl + C n'est donc pas un raccourci pour Copier
 - Utiliser Maj + Ctrl + C à la place
 - Si programme lancé en arrière plan (& en fin de ligne) :
 - Trouver le numéro de processus (pid) avec top
 - 1^{re} col. = PID, dernière col. = nom du programme
 - Tuer le processus avec kill pid

Tuyaux : suite

- Comment ouvrir la GUI du gestionnaire de fichiers à partir du terminal (dans Gnome) ?
 - Appeler le programme par son nom (varie selon la distro) !
 - `nautilus .` (. est le dossier courant)
- Nettoyer le terminal (pour y voir plus clair)
 - `clear`
- `gcc long_nom_de_fichier`
 - taper « `long_` » et appuyer sur TAB
 - Complète automatiquement le nom

Tuyaux : raccourcis claviers (éditeurs)

Fonction	Raccourcis*
Copier la sélection	Cmd + C (C pour Copy)
Couper la sélection	Cmd + X
Coller ce qui a été copié ou coupé	Cmd + V (V est à côté du C)
Annuler une action	Cmd + Z
Tout sélectionner	Cmd + A (A pour All)
Sauvegarder	Cmd + S (S pour Save)
Fermer	Cmd + Q (Q pour Quit)
Imprimer	Cmd + P (P pour Print)
Chercher	Cmd + F (F pour Find)

*Sur les clavier MAC (Historiquement, ça vient de chez Apple)

Tuyaux : raccourcis claviers (éditeurs)

Fonction	Raccourcis
Copier la sélection	Ctrl + C (C pour Copy)
Couper la sélection	Ctrl + X
Coller ce qui a été copié ou coupé	Ctrl + V (V est à côté du C)
Annuler une action	Ctrl + Z
Tout sélectionner	Ctrl + A (A pour All)
Sauvegarder	Ctrl + S (S pour Save)
Fermer	Ctrl + Q (Q pour Quit)
Imprimer	Ctrl + P (P pour Print)
Chercher	Ctrl + F (F pour Find)

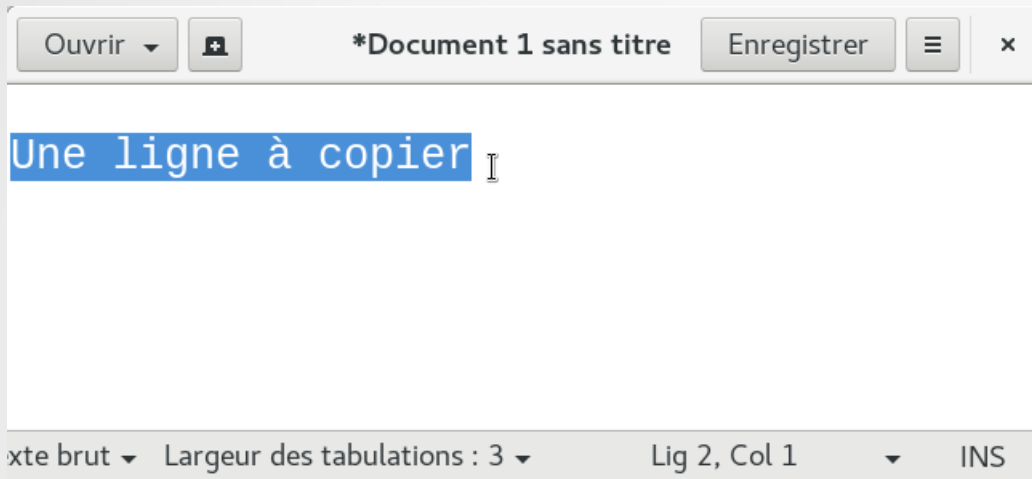
Et sur Windows et GNU/Linux ? Remplacer Cmd par ctrl

Tuyaux : raccourcis claviers (terminal)

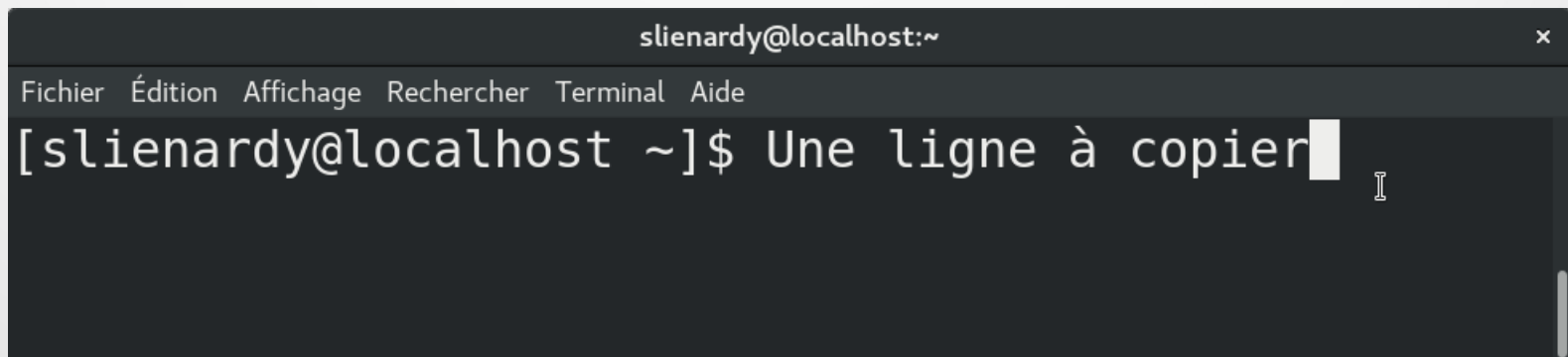
Fonction	Raccourci
Autocomplétion	Tabulation
Interrompre l'exécution	Ctrl + C
Insérer un caractère de fin de fichier (EOF) => permet de fermer l'interpréteur python	Ctrl + D
Retaper la commande précédente	Haut
Retaper la commande suivante	Bas
Tous les fichiers se terminant par .c	*.c
Rechercher TEXTE dans les manpages	/TEXTE
Occurrence suivante de TEXTE	N
Occurrence précédente de TEXTE	Maj + N

One more thing...

- 1) Sélectionner le texte en passant dessus avec la souris et en maintenant le bouton gauche.



- 2) Cliquer sur la molette



- 3) Copié/collé là où se situe le curseur dans la fenêtre active ! Par ex. Dans le Terminal.