

Guide rapide pour utiliser un Terminal

Vous trouverez ci-dessous qqs commandes de base pour démarrer avec votre nouvel environnement sous Linux. Vous trouverez plein de tutoriels en ligne si vous maîtriser l'anglais vous pouvez visiter le site du professeur Maxwell, ou le site pour apprendre le « Shell » : http://linuxcommand.org/lc3_learning_the_shell.php

`cd <répertoire>` - change directory.

Si vous utilisez la commande sans argument, ceci va changer votre répertoire courant avec le répertoire racine. Le répertoire courant est noté comme « . » et le répertoire parent de ce répertoire courant est noté « .. »

- `cd ..` - monter au répertoire parent du répertoire courant.
- `ls` – liste le contenu du répertoire courant
- `pwd` – (print working directory), il donne le répertoire courant
- `mv <from> <to>` - move un fichier, incluant la possibilité de renommer un fichier. La commande `mv` supprimera le fichier de son ancien emplacement après l'avoir copié au nouvel emplacement.
- `cp <from> <to>` - copier un fichier à un nouvel endroit ou nom
- `mkdir <directory name>` - créer un répertoire
- `rmdir <directory name>` - remove (supprimer) un répertoire (il faut que le répertoire soit vide)
- `less <filename>` - il navigue dans un fichier
- `cat <filename>` - il envoie le fichier à la sortie standard
- `echo "a string"` il écrit le txt de la chaîne de caractères à la sortie standard
- `touch <filename>` soit il fait la mise à jour sur un fichier ou il crée un fichier vide si le fichier en question n'existe pas. Ceci peut être pratique dans différents cas, par ex. quand on apprend à créer, renommer et supprimer un fichier via un terminal.
- Piping output: on peut envoyer tout ce qui est envoyé à la sortie standard à la sortie standard via un programme dans un autre programme en utilisant le caractère « | ».
`echo "print 'you are a smart student'" | python`
- Directing output: on peut envoyer des choses à un fichier en utilisant le symbole « plus grand que » « > »
`echo "print 'you are a smarter student'" > smart.py`
- Le caractère « * » est un joker et ça remplace n'importe quelle chaîne de caractères:
`ls c*`

Attention !!!!! Ne faites jamais `rm *` car vous allez tout supprimer !!!

- Tab completion: lorsque vous avez écrit une partie d'un nom de votre fichier ou programme, en tapant sur « tab » complétera automatiquement le reste du nom tant que le choix reste unique. Par exemple si vous avez un seul fichier qui commence par c, tapez c et ensuite « tab » et le nom se complétera automatiquement. Ceci nous permet de gagner du temps !

Si vous avez deux fichiers qui commencent par la même lettre en tapant « tab » vous serez averti du choix possible. (en tapant deux fois, la première fois vous entendrez un bip).

- Taper « Enter » – ce n'est pas nécessaire d'être à la fin de la ligne pour taper return. Si vous allez en arrière et vous faites un changement dans l'écriture de la commande, vous tapez « Enter » et vous exécutez la ligne il faudra l'essayer !
- En tapant « cntl-a » vous allez au début de la ligne, et pour aller à la fin utilisez « cntl-e ». Pour aller en avant utilisez « cntl-f » et en arrière « cntl-b ».

Editeurs de texte

- nano/pico - nano (ou pico dans qqs systèmes) est un éditeur de texte très simple qui marche sur un terminal. Toutes les commandes utilisent la touche « ctrl » et sont visibles en bas, par ex. : <Ctrl x> pour quitter. Pour ouvrir un fichier, il suffit de taper le nom de l'éditeur suivi du nom du fichier.

```
nano <nom_du_fichier>
```

Lorsque vous créez un fichier écrit en python vous pouvez l'ouvrir par exemple `mon_fichier.py` et ajoutez une autre déclaration `print`. Quitter l'application nano et exécutez le fichier en tapant :

```
cat mon_fichier.py | python
```

Ou alors tout simplement :

```
python smart.py
```

- emacs - emacs est un autre éditeur de texte très complet et vous pouvez l'utiliser tout au long de votre vie de programmeur :

Vous le lancez en tapant :

```
emacs mon_fichier.c &
```

emacs est très intelligent ! Dès qu'il voit l'extension il met toutes les librairies nécessaires pour le codage en ce langage ! Le symbole "&" à la fin de la ligne de commande servira à laisser votre session ouverte sur la ligne de commande.

- BBEdit - BBEdit est un éditeur GUI désigné pour écrire du code et du html (web pages).
- JEdit -est un éditeur java multi plateformes et free. Téléchargement possible à partir de jedit.org.