

Programmation I

-

Utiliser la machine virtuelle

Jérôme Kirman

03 octobre 2012

Le terminal

Sous Linux, le meilleur moyen de donner des ordres à l'ordinateur est d'utiliser l'interface en ligne de commande, ou terminal.

Comme l'interpréteur Python, le terminal reçoit des instructions dans un certain langage et les exécute.

Outre des instructions de base (assignation de variables, instructions de contrôle...) qu'on utilisera peu, tous les programmes installés sur la machine sont des instructions.

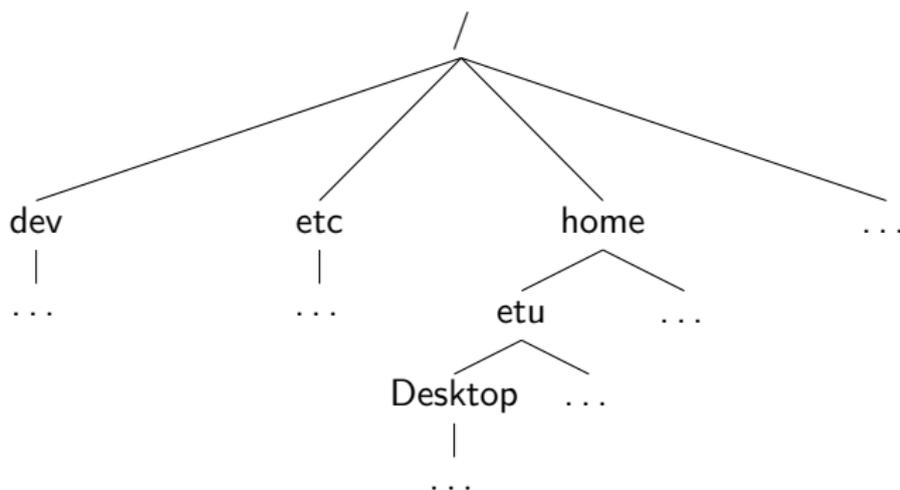
Si une fonction (ou un programme) a besoin d'arguments, ils suivent immédiatement celle-ci sur la ligne, séparés par un espace avant chaque nouveau paramètre.

Comme l'interpréteur Python, le terminal se bloque jusqu'à ce que l'instruction ait été totalement exécutée (= que le programme quitte).

Système de fichiers

Le système d'exploitation (Linux, Windows, ...) travaille sur le disque dur dans un système de fichiers.

Le système de fichiers est une arborescence de répertoires, dans lesquels se trouvent tous les fichiers. On peut le visualiser ainsi :



Une machine virtuelle possède son propre système de fichier (virtuel), en tout point semblable à celui du disque dur réel, mais complètement indépendant.

Répertoire courant et commandes associées

A tout instant, le terminal est actif dans un certain répertoire (dit "répertoire courant").

- La commande `pwd` affiche l'emplacement du répertoire courant.
- La commande `ls` affiche le contenu du répertoire courant.

Depuis un répertoire, on peut accéder facilement à un sous-répertoire (descendre dans l'arbre) ou au répertoire parent (en remontant d'un cran) de façon relative ; il est aussi possible de spécifier un répertoire de manière absolue (avec le chemin complet en partant de la racine).

→ La commande `cd` permet de changer de répertoire. Elle prend en argument le nom du sous-répertoire, `".."` pour le répertoire parent ou encore un chemin complet, relatif ou absolu.

→ Les commandes `mkdir` et `rmdir` permettent de créer ou de supprimer un répertoire (vide) avec le nom spécifié en argument.

Manipulation de fichiers

A tout moment, vous pouvez accéder facilement aux fichiers contenus dans le répertoire courant. Vous pouvez accéder à des fichiers plus lointains en préfixant le chemin pour accéder à leur répertoire.

De nombreuses commandes permettent de manipuler les fichiers, parmi lesquelles :

- `touch` crée un fichier vide avec le nom passé en argument.
- `rm` supprime le fichier passé en argument.
- `cp` copie/colle le fichier donné en premier argument vers le second.
- `mv` fonctionne de la même façon, en supprimant le fichier d'origine.
- `cat` affiche le contenu (sous forme de texte) du fichier spécifié.
- `less` est une version plus évoluée de `cat`. (q pour quitter)

Programmes installés

Les programmes installés sous Linux sont disponibles par une commande :

- `python3.2` - L'interpréteur Python. (exécute un fichier spécifié)
- `geany` - Un éditeur de texte simple. (ouvre un fichier spécifié)
- `firefox` - Le navigateur web de Mozilla. (pour la documentation)
- `dropboxd` - Système de partage Dropbox. (déjà actif par défaut)

Ces programmes peuvent être simplement lancés (sans donner d'argument) ou démarrés en ouvrant un fichier spécifique.

Les commandes (et les programmes) installés possèdent tous une page de manuel qui détaille des options plus avancées. Pour obtenir le manuel d'un programme, on utilise la commande `man` en précisant en argument le nom du programme demandé.

Il existe également une commande `apropos` qui prend en argument un ou plusieurs mots-clés (en anglais) et liste toutes les commandes ayant un rapport avec ces mots clés.

Commandes spéciales

Il existe de nombreuses commandes ou séquences de touches particulières fonctionnant dans un terminal. Parmi les plus utiles à connaître :

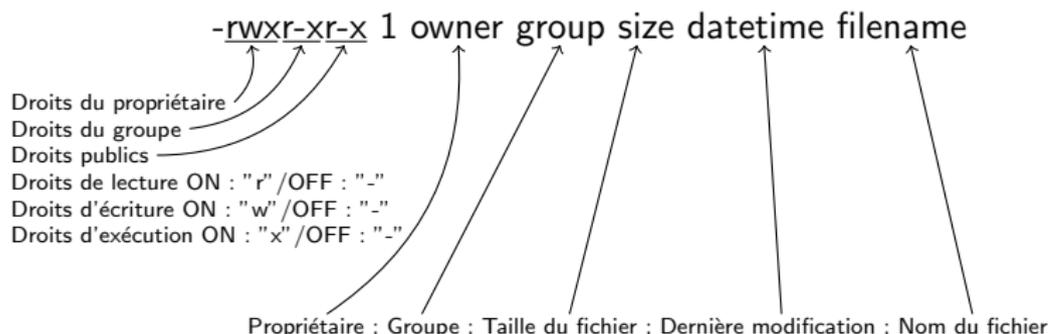
- Ajouter le symbole `&` après une commande l'exécute en arrière-plan : vous pouvez continuer à utiliser le terminal pendant ce temps.
- La séquence `Ctrl` + `C` annule la commande en cours d'exécution.
- La séquence `Ctrl` + `D` envoie un caractère fin de fichier (EOF). Certains programmes, faute d'un fichier précisé en argument, lisent leur contenu directement dans le terminal. Le caractère spécial EOF marque la fin de la lecture (et permet au programme de terminer).

Dernière précision : la philosophie Linux veut qu'un programme qui fonctionne le fasse silencieusement. Si vous avez entré une commande et que rien ne s'affiche, c'est (probablement) que tout s'est bien passé. (Dans le cas contraire, un message d'erreur apparaîtra avec des détails.)

Permissions

Les fichiers et dossiers sous Linux ont des permissions et un propriétaire.

Avec la commande "ls -l", on obtient une liste de lignes de ce style :



L'exécution permet de lancer un programme ou parcourir un répertoire.

`chmod arg1 arg2` change les droits d'accès en `arg1` du fichier `arg2`.

`chown arg1 arg2` donne à l'utilisateur `arg1` le fichier `arg2`.

Autres commandes

Quelques autres commandes Linux usuelles, pour information :

- `ps` - affiche les programmes en cours d'exécution (les processus). Notez que la première colonne contient un chiffre appelé 'PID'.
- `kill` - demande au processus dont le PID est donné en argument de se terminer. Si vous ajoutez l'argument "-9" avant le PID, le système ne prendra pas la peine de demander. (En cas de plantage.)
- `top` - affiche la liste des processus, du plus 'lourd' au plus 'léger'.
- `clear` - 'nettoie' le terminal, pour avoir une vue plus claire.
- `grep` - prend en premier argument un motif, puis un ou plusieurs fichiers. Il affiche toutes les lignes de ces fichiers contenant la séquence de caractères fournie en motif. (Pour chercher du texte.)