

# Premiers pas sous GNU/Linux avec la ligne de commande

Thomas Tempé

Juillet 2001

## Introduction

Aujourd'hui, la plupart des distributions de GNU/Linux viennent avec de jolis programmes d'installation graphique. La configuration d'un système, se fait en quelques clics de souris. Ces progrès indiscutables ont l'inconvénient d'isoler l'utilisateur de la ligne de commande et des longues heures de tâtonnement ô combien didactiques. Le but de ce document est d'introduire les outils merveilleux de notre unix libre aux utilisateurs aventureux qu'une installation en mode texte n'aurait pas rebuté. Il est assumé que le lecteur a un minimum de connaissances en informatique.

Mais pourquoi donc apprendre des outils arcanes pour faire des choses aussi triviales que copier ou éditer des fichiers, alors qu'un glisser-déposer fait la même chose? D'abord, parce que pour peu que vous soyez amené à utiliser un ordinateur quotidiennement, le gain de temps et de confort qu'apporte la ligne de commande par rapport à une interface graphique permet de très vite rentabiliser l'investissement initial de son apprentissage. Mais aussi par passion, parce que l'apprentissage de la ligne de commande est le premier pas vers la découverte de l'un des plus grands piliers de l'informatique : unix.

## 1 Un peu de théorie

On parle souvent de *Linux*. Linux n'est qu'un noyau de système d'exploitation. Il effectue la gestion de la mémoire, implémente les systèmes de fichier, gère la carte son et pilote les disques dur. Mais Linux par lui-même n'est pas bien utile sans un certain nombre d'outils ( librairie C, environnement graphique... ). Le *projet GNU* fournit un grand nombre de ces outils. C'est pour cela qu'il est courant de dire *GNU/Linux* quand on parle du système complet.

Par-dessus le noyau, la librairie C et quelques autres, il est courant de faire tourner un *shell*. Le shell est l'interface utilisateur la plus élémentaire et la plus incontournable qui soit. Un shell est un programme qui permet à l'utilisateur de taper des commandes au clavier. Il les interprète, et les exécute. Il n'est pas plus privilégié qu'un autre programme, et on a couramment un grand nombre de shells ouverts en même temps sur une machine unix. Le shell de prédilection sous Linux s'appelle *bash*, il fait partie du projet GNU.

Chaque shell a son comportement propre. *sh* est le shell de référence, le plus petit dénominateur commun des shells unix. Il est à peine supérieur à la ligne de commande de MS-DOS en terme de confort d'utilisation. Des shells comme *bash* ou *zsh*, en comparaison, sont des outils extrêmement sophistiqués.

Lors du démarrage du système, un certain nombre de distributions sont configurées pour lancer *X-Window* (notez l'absence de "s" final). *X-Window* n'est que la tranche de pain de la tartine : c'est sur lui que tout repose, mais tout seul, il est

un peu fade. Des outils comme KDE, GNOME, Enlightenment ou WindowMaker jouent le rôle de la confiture en proposant un bureau virtuel convivial, des menus, des icônes...

X-Window est une interface graphique client-serveur. Le serveur est la partie qui gère l'affichage et pilote la carte vidéo. Le client exécute les applications, et envoie les requêtes d'affichage au serveur. Les deux s'exécutent typiquement sur la même machine, mais ça n'est pas obligatoire : on peut exécuter un programme sur un ordinateur et l'afficher sur un autre.

L'interface graphique peut sembler nous isoler du besoin d'un shell. C'est une illusion. Toute distribution propose un vaste choix d'émulateurs de terminaux, (`xterm`, `konsole`, `gnome-terminal`, `eterm`...) qui permettent d'utiliser un shell.

## 2 Prise en main

Vous êtes sur un PC avec GNU/Linux. L'interface graphique est démarrée. Appuyez sur **Ctrl-Alt-F1** ( pas **Ctrl-Alt-Suppr!** ). Vous devriez être passé en mode texte. Tapez vos login et mot de passe. Vous êtes sur la console virtuelle numéro 1. Utilisez la commande `tty` pour vous en convaincre.

En maintenant **Alt** enfoncé, naviguez entre les différentes consoles avec les touches de fonction (**F1-F12**) et les flèches gauche et droite. Remarquez en passant comment on peut retourner en mode graphique. Vous pouvez vous loguer sur une autre console, et y exécuter un programme. Toutes les consoles sont indépendantes. Ce mécanisme permet de travailler très efficacement sur plusieurs tâches en parallèle.

## 3 Standing on the Shoulders of Giants

Retour sur la théorie. GNU/Linux est un unix, c'est-à-dire que c'est l'un des nombreux clones de l'Unix originel développé en 1971. Multi-tâche et multi-utilisateur, Unix a été conçu à une époque où il était impensable d'avoir un ordinateur par personne. Il faut donc voir votre machine comme une sorte de serveur. Toute personne autorisée peut y accéder par l'intermédiaire du réseau, et y lancer des programmes comme s'il était assis devant. Plusieurs personnes peuvent s'en servir simultanément, de manière transparente.

Un unix n'appartient pas à son utilisateur. L'utilisateur est un invité parmi d'autres. Il n'a comme droits que ceux qu'on lui accorde. Le système appartient à l'administrateur, appelé *root*.

**Règle numéro 1** : Ne jamais se loguer en tant que *root*. C'est inutile et dangereux. Quand une action nécessite les privilèges de *root*, utiliser la commande `su` pour l'exécuter.

Bien sur, ce genre de règles, c'est comme les copies de sauvegarde. Personne ne les applique jamais avant d'avoir fait : `rm -rf /bin`. Unix, c'est avant tout une affaire d'expérience :-).

## 4 Le système de fichiers

Parlons maintenant de la façon dont est organisé un système de fichiers. Au commencement est la racine ( / ). La racine appartient à l'OS. Tapez `ls /` La racine contient un certain nombre de répertoires dont le sens est gravé dans la tradition ancestrale de unix. Le répertoire `/bin`, par exemple, contient les fichiers exécutables (binaires) essentiels au système.

Le répertoire `/home` contient un sous-répertoire pour chaque utilisateur. C'est là où sont créés tous vos fichiers personnels, et où vous vous trouvez quand vous ouvrez un shell. Tapez `pwd` (Print Working Directory) pour vous en convaincre. Le répertoire personnel est représenté par le chemin `~/`. `~/lettres_d'amour` est la même chose que `/home/<nom d'utilisateur>/lettres_d'amour`. Tapez `ls -l /home` pour avoir un aperçu de ce qu'il vous reste à apprendre sur les attributs des fichiers (commandes `chown`, `chmod`, `chgrp`...)

`/usr` contient les programmes destinés aux utilisateurs, `/var` des fichiers de données changeant, mais qui n'appartiennent à aucun utilisateur en particulier (logs, courrier, bases de données, etc...), et `/mnt` contient les point de montage de vos différents périphériques.

Pour accéder au contenu d'une disquette ou d'un CD, on le projette à l'intérieur d'un répertoire, ou point de montage, à l'aide des commandes `mount` et `umount`. Le montage et le démontage se font automatiquement dans certaines distributions.

Pour les utilisateurs de DOS : les différences les plus flagrantes a priori sont le fait que Linux fait la différence entre majuscules et minuscules, et qu'il faut mettre un espace entre `cd` et `...`. Essayez aussi `pushd` et `popd`. `alias` permet de faire une première personnalisation, par exemple en ajoutant la ligne `alias ls=pushd` dans votre fichier `~/bashrc`.

Notez que le nom des fichiers cachés sous unix commence par un point, et qu'on les affiche avec `ls -a`. Les paramètres de la plupart des commandes unix peuvent être agglomérés, dans la mesure où cela n'introduit pas d'ambiguïté. Par exemple, `ls -a -l` est équivalent à `ls -al`.

Comme sous DOS, la variable d'environnement `PATH` contient la liste des répertoires où le shell cherche les programmes exécutables. Exécutez `export`, ou `echo $PATH` pour l'afficher. Notez bien que `PATH` représente le nom de la variable, et `$PATH` son contenu. `export PATH=$PATH :~/bin` permet d'ajouter vos exécutables personnels dans votre home. Si vous souhaitez exécuter un programme qui ne se trouve pas dans les répertoires du `PATH`, tapez son nom précédé d'un `./`. Il s'agit d'un mécanisme permettant d'éviter d'exécuter involontairement un fichier d'origine inconnue.

## 5 Dieu créa le fichier texte.

### Tout le reste est l'oeuvre de l'homme.

Votre meilleur ami ici bas est un éditeur de texte. Je me permets de conseiller *GNU Emacs*, d'autres préféreront *Vim*. C'est une affaire de personnalité. Inutile d'essayer *gedit* ou *kwrite*. Ce sont des outils primitifs, sans aucune sophistication. Un utilisateur expérimenté de *gedit* saura tout juste économiser quelques clics pour sauver un fichier, alors qu'un utilisateur expérimenté de Emacs pourra éditer son code presque sans contraintes, par le seul pouvoir de sa pensée.

Le répertoire `/etc` contient les fichiers de configuration de votre système unix. Ce sont tous des fichiers texte. Tapez `emacs /etc/bash.bashrc`. Notez au passage comme vous pouvez vous éviter de saisir toute la commande en utilisant la touche de tabulation.

Le fichier `/etc/bash.bashrc` est un script shell qui s'exécute à chaque ouverture d'un shell, pour tous les utilisateurs. Notez que si vous êtes arrivé jusqu'à là, c'est que vous n'êtes pas root, et que le fichier est accessible en lecture seule (ce qui est indiqué par les petits symboles `%` en bas à gauche de votre écran). Explorez-le un petit peu. Vous verrez qu'il contient des commentaires et des lignes de programme. Il y a aussi souvent le nom de l'auteur et son adresse e-mail. Ceci est très important, car il s'agit de logiciel libre.

Pour quitter `emacs`, vous pouvez utiliser **Ctrl-x Ctrl-c**, et pour enregistrer vos modifications, utilisez **Ctrl-x Ctrl-s**.

## 6 Les filtres

Loguez-vous dans une autre console, et tapez `echo "Hello world"`. `echo` est une commande qui prend un argument, et l'envoie sur la *sortie standard*. La sortie standard est redirigée vers l'écran. Vous devriez avoir vu quelque chose. (Notez au passage qu'il est nécessaire de mettre des guillemets autour d'un argument contenant un caractère d'espacement ou caractère spécial. On aurait aussi pu écrire `echo Hello\ world`).

Tapez maintenant `echo "Hello world" > FooBar`. Vous n'avez rien vu? Normal : la sortie standard de `echo` a été redirigée vers le fichier `FooBar` du répertoire courant. Tapez `cat FooBar`. `cat` envoie le contenu d'un fichier sur la sortie standard, qui est votre écran.

La ligne `echo "alias e=emacs" >> ~/.bashrc` ajoute la commande `alias e=emacs` à la fin de votre fichier `.bashrc`, est exécuté à chaque fois que vous démarrez un shell `bash`. Cette commande vous permet d'abrégier le lancement de `emacs`. Vous pouvez l'essayer immédiatement en la saisissant dans votre shell. Attention, si vous aviez tapé : `echo "alias e=emacs" > ~/.bashrc`, vous auriez **remplacé** votre fichier par cette seule commande.

`less` est un afficheur page par page. Il affiche soit le fichier dont le nom lui est donné à la ligne de commande ( `less FooBar` ), soit le contenu de son entrée standard. Pour rediriger la sortie standard de `cat` vers l'entrée standard de `less`, tapez `cat FooBar | less`. (pour quitter `less`, tapez `q`).

Ce mécanisme est particulièrement intéressant, mais nécessite un peu d'entraînement. Par exemple, en sachant que la commande `grep` filtre un fichier ou son entrée standard et n'affiche que les lignes contenant un mot donné, et que `ps` écrit sur la sortie standard la liste des processus que l'on est actuellement en train d'exécuter, il est possible de taper `ps | grep bash` pour retrouver rapidement le numéro de processus de `bash`. Petit exercice : trouvez ce que fait la ligne suivante : `ps -ax | grep bash | grep -v grep | wc -l`

Autre exemple : compressez le fichier `FooBar` à l'aide de la commande `gzip FooBar`. Un appel à `ls` vous indiquera que `FooBar` a été remplacé par `FooBar.gz`. Pour le relire page par page, trois possibilités : `gunzip FooBar.gz ; less FooBar`, `zcat FooBar | less` ou `zless FooBar`.

Vous pouvez maintenant supprimer tous les fichiers commençant par `FooBar`, à l'aide de la commande `rm FooBar*`.

## 7 Les processus

Lancez la commande `cat`, sans argument. Celle-ci attend du texte sur son entrée standard (le clavier) et le reproduit sur sa sortie standard. Tapez quelques lignes suivies de `<entrée>`.

Pour terminer `cat`, deux solutions : **Ctrl-C** tue le processus sans pitié, et **Ctrl-D** le termine proprement, en lui signalant une **fin de fichier** sur l'entrée standard.

Relancez `cat`, et allez dans une autre console. Tapez `ps`. `ps` liste les processus actifs dans cette console. Tapez `ps -a`, qui liste tous les processus d'un utilisateur. Vous pourrez y repérer `cat`, et trouver son numéro de processus. Pour le chercher plus facilement, essayez `ps ax | grep cat`. La commande `grep` filtre toutes les lignes de l'entrée standard (ou d'un fichier), et ne laisse passer que celles qui

contiennent le mot-clé donné. Pour tout savoir sur les processus s'exécutant sur votre machine, essayez `ps aux`.

Vous avez repéré le numéro de processus (ou PID) de `cat`. Tuez-le maintenant avec la commande `kill <PID>`. Le même travail peut être accompli à l'aide de `killall cat`. Les processus récalcitrants peuvent être tués à l'aide de `kill -9 <PID>`. Lisez le manuel de `signal` pour en savoir plus.

Lancez maintenant Emacs, et tapez `Ctrl-z`. Emacs est interrompu, mais continue d'exister en tâche de fond. Convainquez-vous-en à l'aide de la commande `ps`. Quand un processus est interrompu, il ne peut plus rien faire, il est bloqué. Pour lui laisser reprendre son fonctionnement de tâche de fond, utilisez `bg` (*background*). Pour le faire revenir en avant-plan et continuer votre travail, essayez `fg` (*foreground*).

## 8 RTFM : Read The Fine Manual

Loguez-vous sur une console virtuelle voisine, et tapez `man inittab`. De la bonne littérature. Une des blagues qui entoure unix est que les experts gardent leur monopole sur la connaissance en assénant aux néophytes de grands coups de manuels sur la tête. Au moins, un linuxien n'a pas d'excuse pour ne rien apprendre.

Tapez maintenant `man ls`, et comptez le nombre de pages disponibles, histoire de vous convaincre de la quantité de documentation disponible, et de la richesse des outils GNU. Utilisez le raccourci `/` pour chercher le mot `sort`, histoire de voir si on peut trier les fichiers selon leur taille. Notez également la section *Voir aussi* en fin de `man`.

Tapez `ls --help`, puis `info ls`. Info est un système de documentation hypertexte. Vous pouvez y naviguer à l'aide des touches `n` (*next*), `p` (*previous*) et `u` (*up*). Le lecteur `info` s'utilise comme Emacs, que vous devriez maintenant savoir quitter.

Les *man pages* ou manuels (`man ls`) sont des documents de référence. Assez indigestes comme livres de chevets, ils permettent de retrouver une information ou découvrir une commande rapidement. La plupart des programmes acceptent un paramètre `--help` qui donne un résumé très succinct des paramètres acceptés. Info agit souvent comme une sorte de super-man pour les programmes les plus volumineux.

Chaque paquet installé sur le système contient sa propre documentation. Celle-ci est souvent localisée dans `/usr/share/doc/<nom du paquet>`. (Voyez comme la complétion automatique de Bash avec la touche de tabulation peut être utile). Allez voir le fichier README de la documentation du programme `mc` à l'aide de la commande `less`. Profitez-en pour essayer la différence entre `less`, `more`, `cat`, `head` et `tail`. Notez que `mc` est souvent le meilleur ami des débutants sous GNU/Linux.

## 9 Il faut déjà savoir beaucoup de choses avant de pouvoir demander ce qu'on ne sait pas

*Lynx* et *Links* sont deux navigateurs Web en mode texte. Servez-vous-en pour aller lire la documentation du paquet HOWTO. Jetez un oeil à la liste des HOWTO disponibles, et regardez-en un ou deux (utilisez les touches de direction pour naviguer, et `Q` pour quitter). Ces docs sont une mine d'or pour le linuxien désœuvré : rien de plus instructif pour tuer le temps que d'aller lire un HOWTO. Rien de plus efficace, surtout, pour apprendre comment marche le réseau sous Linux ou tout savoir sur la gravure de CD. Les HOWTO constituent une partie importante de l'arsenal de documentation disponible, et une source insondée de connaissances.

L'apprentissage de GNU/Linux ne s'arrête jamais, pour peu qu'on aie commencé un jour. Impossible de tout couvrir, mais j'espère avoir donné quelques indices pour

démarrer. Bien sûr, si le mode texte vous rebute, rien ne vous empêche de travailler dans une fenêtre de terminal en mode graphique ; mais une chose est sûre, c'est qu'aucune autre interface utilisateur n'arrive encore à la cheville de la ligne de commande quand il s'agit de prendre en main son système.

La suite des opérations passera probablement par une exploration des outils de la ligne de commande (à grands coups de manuels), parmi lesquels `ln`, `mkdir`, `find` ou `file`, `nice`, et tous ceux abordés dans cette doc.

De même, il est indispensable d'examiner plus en profondeur le système de fichiers, et particulièrement le contenu de `/etc`. Il faudra aussi apprendre un éditeur de texte (Pour Emacs, de la documentation est disponible sur <http://gpl.insa-lyon.fr>).

Certaines distributions de GNU/Linux sont particulièrement propices à l'apprentissage, comme la *Debian*, la *Gentoo* ou la *Slackware*. D'autres unix libres existent également, parmi eux *OpenBSD*.

## 10 Documentation

Si vous avez lu cette documentation jusqu'ici, félicitations. Vous êtes maintenant prêt à vous lancer dans le grand vide.

Comme elle contient beaucoup d'informations, ne vous attendez pas à tout comprendre ou retenir immédiatement. Gardez-la sur un coin de bureau, et revenez-y de temps en temps.

Sur Internet, les sites les plus incontournables sont :

- <http://www.lea-linux.org> pour la documentation en français,
- <http://www.linuxfr.org> pour se tenir au courant,
- <http://freshmeat.net> pour chercher un programme sous Linux,
- <http://www.sourceforge.net> si vous voulez développer,
- <http://www.gnu.org> pour sa section 'Philosophie' que tout linuxien se doit d'avoir lue,
- <http://www.refcards.com/about/emacs.html> pour apprendre Emacs, et éventuellement
- <http://www.slashdot.org> pour perdre son temps.

Ceux qui souhaitent trouver une documentation du même goût, mais plus complète peuvent se référer à *Le système Linux*, de Matt Welsh, Matthias Kalle Dalheimer et Lar Kaufman, paru aux éditions O'Reilly (ISBN : 2-84177-086-9). Il est disponible dans toutes les bonnes bibliothèques techniques ( Doc'INSA, bibliothèques de départements).

Enfin, l'adhérence à un *LUG* (groupe d'utilisateurs de Linux) est une source importante d'informations.