

# Découverte de GNU/Linux

Arnaud Bannier  
Nicolas Bodin  
Aurélien Texier

## 1. Avant c'était tout noir !

Parce qu'il y a quelques années encore, l'informatique n'était pas ce que l'on connaît aujourd'hui, piqûre de rappel pour tout le monde. Nous sommes en 1985, l'informatique est ouvert depuis peu au grand public. Pour avoir un aperçu de l'histoire, nous allons réaliser les exercices suivants en utilisant la *ligne de commande*. Cela permet d'utiliser l'ordinateur sans la souris et donc sans se fatiguer (c'est que vous diront les geeks).

### 1.1. Se connecter

Lors du démarrage de l'ordinateur, restez attentif pour choisir de démarrer sous Ubuntu et non sous Windows qui est le choix par défaut. On vous demande alors votre identifiant et mot de passe. Tapez la combinaison de touches (i.e. appuyer sur toutes les touches en même temps) :

`ctrl-alt-f2 .`

L'écran devient alors noir, vous arrivez sur l'un des nombreux terminaux de linux. Le système attend que vous donniez votre identifiant ou `login` (saisissez-le précédé du domaine `etd-1\` puis appuyez sur `entrée`) puis votre mot de passe ou `password` (tapez-le en respectant les majuscules et les minuscules puis appuyez sur `entrée`). Rien ne s'affiche lors de l'entrée du mot de passe, c'est normal, donc attention à bien le saisir. De plus, votre mot de passe est strictement personnel (comme le numéro d'une carte bancaire) et engage votre responsabilité, donc conservez-le secret.

Imaginez maintenant ce terminal comme un explorateur de documents. Vous vous situez dans un dossier particulier et vous pouvez réaliser des actions sur ce dossier telles que ajouter un nouveau document, en supprimer un, lister l'ensemble des fichiers de ce dossier etc. Au lieu de réaliser un clic droit dans la fenêtre, ces actions sont accessibles grâce à un ensemble de *commandes*. Le tableau 1 liste les principales commandes à connaître<sup>1</sup>.

Notez enfin l'existence de deux dossiers magiques notés « `.` » et « `..` ». Le point seul indique le chemin relatif du répertoire actuel. S'il ne vous semble par particulièrement utile pour le moment, vous verrez qu'il nous sera très rapidement utile. Les deux points indiquent le chemin relatif vers le répertoire parent, ce qui est très pratique pour accéder rapidement à un contenu de ce répertoire.

---

1. Une liste bien plus complète peut se trouver sur Wikipédia : [https://fr.wikipedia.org/wiki/Commandes\\_Unix](https://fr.wikipedia.org/wiki/Commandes_Unix)

Nom <b>arg</b>	Utilité
<code>pwd</code>	Affiche le chemin absolu du répertoire courant
<code>mkdir Rep</code>	Crée le dossier <b>Rep</b> dans le répertoire courant
<code>rmdir Rep</code>	Supprime le répertoire <b>Rep</b>
<code>cd Rep</code>	Se déplacer dans le répertoire <b>Rep</b>
<code>ls</code>	Liste le contenu du répertoire actuel
<code>chmod</code>	Change les droits d'un fichier
<code>chown</code>	Change le propriétaire d'un fichier
<code>touch File</code>	Crée le fichier <b>File</b> dans le répertoire courant
<code>rm File</code>	Supprime le fichier <b>File</b>
<code>cp src dst</code>	Copie le fichier <b>src</b> à l'emplacement <b>dst</b>
<code>mv src dst</code>	Déplace le fichier <b>src</b> vers l'emplacement <b>dst</b>

TABLE 1 – Les principales commandes unix

## 1.2. Manipulations de base

Lorsque vous vous connectez, vous êtes à la racine de votre compte. Quel est le chemin absolu de ce répertoire ?

Combien existe-t-il de fichiers dans ce répertoire d'accueil ?

À la racine de votre compte, créez les répertoires **SRC**, **EXE** et **BIDON**. Quelle commande supprime le répertoire **BIDON** ?

Maintenant, descendez dans le répertoire **SRC** et créez un fichier vide **empty.txt**. Quelles commande utilisez-vous ?

## 1.3. Le manuel

La commande ultime sous Linux reste l'appel au manuel qui se réalise en tapant **man** suivi du nom d'une commande. Qu'il soit en français ou en anglais, le formatage est identique. Une *page de man* se compose de plusieurs champs dont les trois premiers sont le nom de la commande (contenant un résumé de la commande), le synopsis permettant d'un seul coup d'œil de voir l'ensemble de options disponibles, et la description contenant l'explication de chaque option.

Quelles sont les options de la commande `pwd` ?

Combien existe-t-il de sections de `man` ?

## 2. Un peu de couleur

L'histoire c'est sympa mais c'est un peu vieux jeu. Revenons dans le présent et utilisons l'interface actuelle. Déconnectez-vous du terminal actuel en tapant `logout` puis tapez la combinaison de touches :

`ctrl-alt-f7` .

Entrez votre login et votre mot de passe. Finalement, Linux n'est pas si différent de Windows, n'est-ce pas ? Familiarisez-vous avec cette nouvelle interface, repérez les icônes et raccourcis utiles (Firefox, gestion du système, Documents ...). Dans la barre de recherche, tapez le mot **Terminal**. Vous venez d'ouvrir l'équivalent du mode console vu précédemment, tout en conservant derrière la convivialité de l'interface graphique.

### 2.1. Équivalence du mode graphique et du mode console

Tapez à nouveau dans la console la commande permettant de vous situer. Dans **Raccourcis** → **Documents**, trouvez le dossier indiqué par la console. Mettez la fenêtre de navigation à côté de la console pour avoir les deux en vue simultanée.

Créez un dossier nommé **TEST** à l'aide de la console et observez ce qui se passe dans le navigateur. À l'aide de la fenêtre de navigation, créez un dossier appelé **Mon Dossier** puis utilisez la console pour y descendre.

Comment contourner le problème de l'espace dans le nom du dossier ?

### 2.2. Les éditeurs de texte

Maintenant que vous savez vous déplacer, vous repérer et supprimer des fichiers et documents, il est temps d'apprendre à créer des fichiers ! Plusieurs moyens s'offrent à vous.

**L'éditeur bureautique (type traitement de texte) :** il est principalement utilisé pour... du traitement de texte bien sûr, donc non adapté au codage. La raison est que le texte écrit est formaté dans un type précis, tout comme la mise en page etc. Ces informations n'apparaissent pas lorsqu'on ouvre le document mais sont présentes dans le fichier. Pour preuve regardez la taille d'un document bureautique contenant uniquement quelques mots...

**L'éditeur de texte (type emacs, gedit) :** le fichier contenant le texte entré est directement codé suivant la table ASCII (1 caractère = 1 code sur 1 octet). N'importe quel éditeur de texte brut sur des OS différents peut donc afficher le texte contenu dans le fichier.

**L'éditeur en ligne de commande (type vim, nano) :** il possède les mêmes fonctions que l'éditeur de texte mais n'est pas en mode graphique. Tout se passe dans la console et les actions telles que la sauvegarde et la modification sont réalisées par le biais d'une combinaison de touches (**esc : w** pour sauvegarder sous vim par exemple). L'avantage principal est sa légèreté et peut donc être implanté dans des environnements disposant de peu de ressources.

**l'IDE (Environnement de Développement Intégré type Code::Blocks) :** il combine l'éditeur de texte, le compilateur, le debugger etc. Il est donc ultra-complet et permet de gérer facilement des gros projets composés de plusieurs dizaines de fichiers.

Pour le moment, nous allons coder nos programmes dans un, voir deux fichiers. Utiliser un IDE est donc inutile et nécessite tout de même une prise en main pas nécessairement intuitive. Reste donc **vim**, **gedit** et **emacs**. Libre à vous d'utiliser celui que vous préférez, nous préconisons tout de même l'emploi de **gedit** pour plus de simplicité.

Ouvrez l'éditeur de texte pour créer un fichier nommé **bonjour.c** puis tapez en respectant scrupuleusement la syntaxe et la mise en page le code suivant :

```
1  #include <stdio.h>
2
3  main()
4  {
5      printf("Ceci est mon premier programme.\n");
6  }
```

Enregistrez ce fichier dans le dossier **SRC** créé précédemment. Trouvez, à l'aide de la console et de la commande **man**, sa taille en **ko**.

## 2.3. Les fichiers

Quelle commande copie le fichier `bonjour.c` directement à la racine de votre compte ?

Sachant que « `..` » représente le répertoire parent, quelle commande déplace le fichier `bonjour.c` de la racine de votre compte dans le répertoire `TEST`.

Quelle commande permet de renommer le fichier `bonjour.c` en `main.c` ?

Comment supprimer le fichier `main.c` ?

## 2.4. Les droits

Linux est fondé sur un système droits : chaque utilisateur de la machine est référencé dans un groupe et possède certains droits sur les fichiers concernant leur lecture, écriture ou exécution. L'option `-l` de la commande `ls` permet de visualiser en début de ligne les droits associés à ce fichier. Pour chaque fichier et dossier, une ligne telle que ci-dessous apparaît :

```
drwxr-xr-x 16 util groupe 4096 2012-09-19 11:18 Documents
```

Voici le détails des champs :

**drwxr-xr-x** : 10 caractères présentant le type et les droits de l'élément. Le premier caractère indique le type de l'élément : nous avons donc ici un dossier (**d**), **-** dans le cas d'un fichier. Les trois caractères suivants indique les droits en lecture (**r**), écriture (**w**) et exécution (**x**) de l'utilisateur, un **-** indiquant que les droits sont absents, les trois suivants du groupe et les trois derniers des autres personnes.

**16 util groupe** : **16** indique le nombre de liens physiques associés à cet élément (il a 16 endroits différents dans l'ordinateur permettant d'accéder à **Documents**). **util** indique le nom de l'utilisateur possédant cet élément, **groupe** indique le nom du groupe de l'utilisateur.

**4096** : indique la taille de l'élément en blocs mémoire. Ce ne sont pas des octets !

**2012-09-19 11:18** : date de dernier accès.

**Documents** : nom de l'élément.

La commande `chmod` permet de changer les droits affectés à l'élément en utilisant les options `[ugoa] [+ -=] [rwx]`. Allez maintenant dans le répertoire `SRC`. Quelles sont les commandes pour :

- ajouter tous les droits à l'utilisateur ;
- supprimer les droits d'exécution et d'écriture au groupe ;
- que le reste du monde ne puisse ni lire, ni écrire ni exécuter le fichier.

Aidez vous des commandes `man`, `chmod` et `ls`.

On peut également attribuer un nombre à un élément indiquant ses droits : Le nombre possède alors trois chiffres, le premier pour `u`, le deuxième pour `g` et le dernier pour `o`. Chaque chiffre est une somme de puissance de 2. Le 1 représente les droits en exécution, 2 pour l'écriture et 4 pour la lecture. Ainsi pour accorder les droits utilisateur (`rw`), groupe (`r-x`) et autres (`--`) on écrit `chmod 750 <fichier>`. Donnez donc les commandes correspondantes pour réaliser les actions données précédemment.

### 3. Trucs & astuces

Ces questions vont nécessiter une recherche de votre part sur Internet. Le but est de vous faire découvrir des astuces pour vous faciliter la vie avec la console et les programmes disponibles sur Linux.

Vous pouvez lancer un programme simplement en entrant son nom dans la console, mais celle-ci semble alors verrouillée (les commandes que vous entrez n'ont pas d'effet tant que le programme est lancée). Il existe un symbole qui permet de lancer votre programme sans bloquer votre console, lequel est-ce ?

On l'a vu plus haut, la commande `rmdir` permet de supprimer un répertoire, mais uniquement s'il est vide. Est-il possible de le supprimer même s'il n'est pas vide ?

La commande `grep` vous permet de trouver un **pattern** dans un texte et vous l'affiche. Je possède un dossier contenant de nombreux fichiers, et je souhaite trouver ceux dont le nom comporte le mot « image ». Il existe pour cela le symbole « *pipe* (tuyau en français) » | (`AltGr + 6`) qui permet de combiner des commandes. Comment puis-je l'utiliser dans mon cas ?

À l'aide de la commande `ls` et d'une astuce, je souhaite afficher toutes les images JPEG de mon dossier. Quelle est cette astuce ?

Je ne trouve pas mon fichier `toto.txt` sur mon `/home` ! Quelle commande peut m'aider ?

Je dois ajouter la date sous la forme `jj/mm/aaaa` à la fin de mon fichier `toto.txt`. Comment faire ?

---

## 4. Sauvegarde des données

Toutes les données que vous avez créées lors de ce TD sont pour le moment stockées *en local* sur la machine que vous venez d'utiliser et seront supprimées au moment de votre déconnexion. Il est donc important de les sauvegarder sur votre espace en ligne. Pour cela, à l'aide d'un explorateur de documents, cliquez sur le dossier `montage-lully` situé dans le bandeau gauche de la fenêtre. Saisissez les premières lettres de votre login afin de trouver votre dossier personnel. Référez-vous à la figure 1, page suivante.

Vous arrivez directement sur votre espace en ligne dont vous seul avez accès. Déplacez les fichiers créés lors du TD vers ce répertoire afin d'en conserver une trace. N'oubliez pas de verrouiller votre session à l'aide de la combinaison de touches `ctrl-alt-L` lorsque vous quittez votre poste temporairement. Déconnectez vous de votre session à la fin de votre TD.

Lors de chaque TD/TP, nous vous demandons de travailler en local sur votre machine, c'est à dire que la rédaction d'un programme, sa compilation et son exécution **ne doivent pas** se faire sur votre compte en ligne mais plutôt sur votre machine locale.

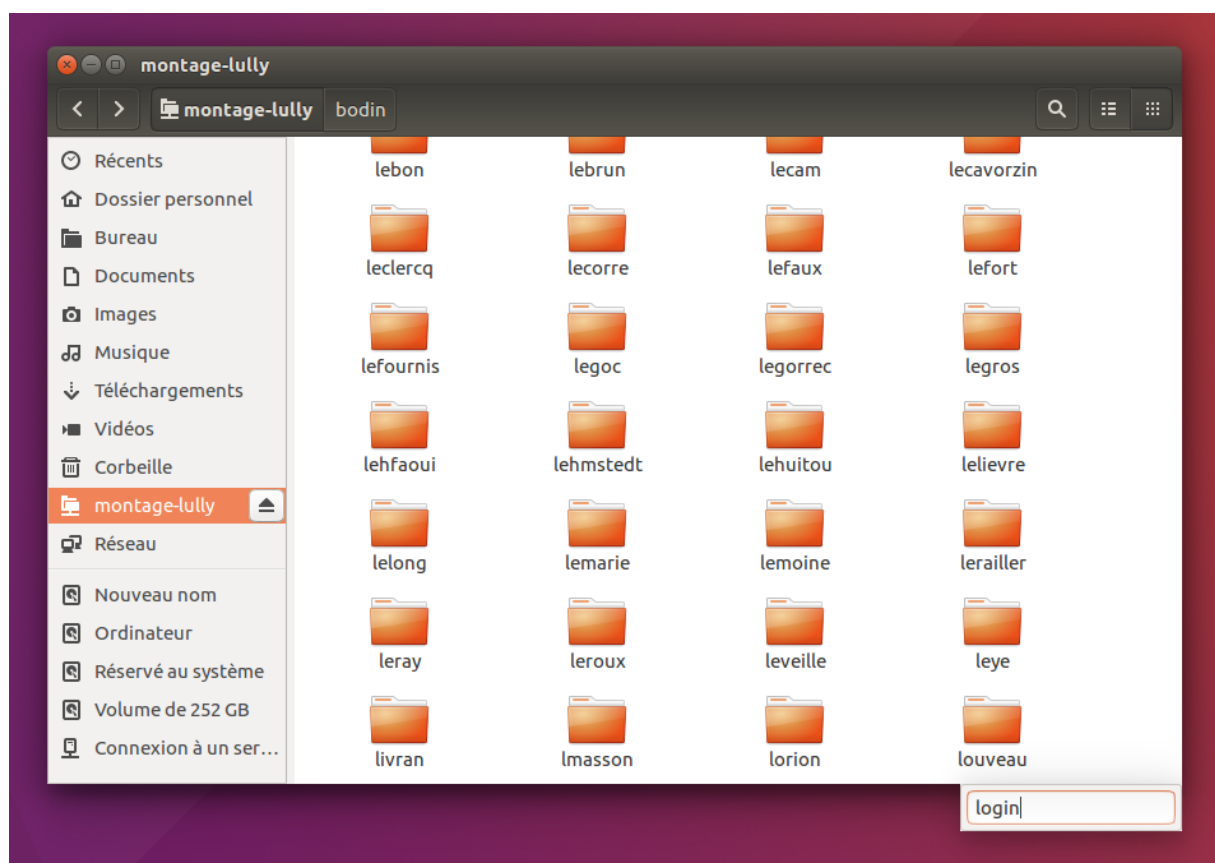


FIGURE 1 – Sauvegarde des données