

Linux

Linux est une famille entière de systèmes d'exploitation Unix à source ouverte basés sur le noyau Linux. Cela inclut tous les systèmes les plus populaires basés sur Linux comme **Ubuntu**, **Fedora**, **Mint**, **Debian**, et d'autres. Plus précisément, on les appelle des distributions ou des distros.

Créé par Linus Torvalds



- Chaque personne qui utilise une machine Linux possède un compte utilisateur.
- Les utilisateurs sont classés par groupes.
- Il existe un **superutilisateur** qui a tous les droits : **root**. C'est l'administrateur de la machine, le seul à être autorisé à installer des programmes ou effectuer certaines modifications sur le système.
- Certaines commandes ne fonctionnent que lorsqu'on est root et nécessitent donc de se transformer en root à l'aide de `sudo` (voir #11). C'est le cas de la commande d'ajout d'utilisateur (`adduser`), de suppression d'utilisateur (`deluser`)

Les raccourcis à connaître pour accéder à la console en plein écran :

- `Ctrl + Alt + F1` : terminal 1 (tty1) ;
- `Ctrl + Alt + F2` : terminal 2 (tty2) ;
- `Ctrl + Alt + F3` : terminal 3 (tty3) ;
- `Ctrl + Alt + F4` : terminal 4 (tty4) ;
- `Ctrl + Alt + F5` : terminal 5 (tty5) ;
- `Ctrl + Alt + F6` : terminal 6 (tty6) ;
- `Ctrl + Alt + F7` : retour au mode graphique.

« Terminal » est un autre nom pour « Console ».

Sous Unity, rendez-vous dans le lanceur et recherchez "Terminal"

0. apt-get

Outil de ligne de commande pour la gestion des paquets et de leurs dépendances.

Tous les paquets sont regroupés au sein d'un même serveur appelé dépôt (*repository* en anglais).

En fait il existe un grand nombre de dépôts. La plupart proposent exactement les mêmes paquets (les dépôts sont donc des copies les uns des autres).

Certains dépôts spéciaux proposent toutefois des programmes que l'on ne trouve nulle part ailleurs, mais il est rare que l'on ait besoin de se servir dans ces dépôts.

On télécharge les programmes depuis des serveurs (fournis par Ubuntu, votre fournisseur d'accès ou une université) qui font office de dépôts.

En France, par défaut, Ubuntu utilise le dépôt : <https://fr.archive.ubuntu.com>

- **paquet** : c'est un programme « prêt à l'emploi », l'équivalent des programmes (.exe) d'installation sous Windows en quelque sorte ;
- **dépendance** : un paquet peut avoir besoin de plusieurs autres paquets pour fonctionner, on dit qu'il a des dépendances ;
- **dépôt** : c'est le serveur sur lequel on va télécharger nos paquets.

Gérer les dépôts

```
sudo nano /etc/apt/sources.list
```

Vous pouvez utiliser apt-get en tant que root [**s**uper **u**tilisateur **su**] (voir #11)

Ex : sudo apt-get install

- **apt-cache search *nom-paquet*** : effectue une recherche de paquet dans votre cache
- **apt-get install** : télécharge et installe un paquet
- **apt-get remove** : désinstaller un paquet
- **apt-get remove** : désinstaller un paquet et ses dépendances
- **apt-get update** : mise à jour du cache des paquets
- **apt-get upgrade** : mettre à jour les paquets après avoir mis à jour le cache (update)
- **apt-get dist-upgrade** : mettre à jour les paquets avec résolution de conflits

1. Commande pwd

Utilisez la commande **pwd** pour trouver le chemin du répertoire de travail (dossier) dans lequel vous êtes actuellement. La commande retournera un chemin absolu (complet), qui est en fait un chemin de tous les répertoires qui commence par une barre oblique (/).

Un exemple de chemin absolu est **/home/utilisateur**.

2. Commande cd

Pour naviguer dans les fichiers et répertoires de [Linux](#), utilisez la commande **cd**. Elle nécessite soit le chemin d'accès complet, soit le nom du répertoire, selon le répertoire de travail dans lequel vous vous trouvez.

Disons que vous êtes dans **/home/utilisateur/Documents** et que vous voulez aller dans **Photos**, un sous-répertoire de **Documents**. Pour ce faire, il vous suffit de taper la commande suivante : **cd Photos**.

Un autre scénario est possible si vous voulez passer à un répertoire complètement nouveau, par exemple, **/home/utilisateur/Films**. Dans ce cas, vous devez taper **cd** suivi du chemin absolu du répertoire : **cd /home/utilisateur/Films**.

Il existe quelques raccourcis pour vous aider à naviguer rapidement :

- **cd ..** (avec deux points) pour se déplacer d'un répertoire vers le haut
- **cd** pour aller directement au dossier principal (home)
- **cd-** (avec un tiret) pour passer à votre répertoire précédent
- **cd ~** retour à la racine

Par ailleurs, le shell de [Linux](#) est sensible à la casse. Vous devez donc taper les noms des répertoires exactement comme ils sont.

3. Commande ls

La commande **ls** est utilisée pour visualiser le contenu d'un répertoire. Par défaut, cette commande affichera le contenu de votre répertoire de travail actuel.

Si vous voulez voir le contenu d'autres répertoires, tapez **ls** et ensuite le chemin d'accès du répertoire. Par exemple, tapez **ls /home/utilisateur/Documents** pour voir le contenu de **Documents**.

Il existe des variantes que vous pouvez utiliser avec la commande **ls** :

- **ls -R** énumérera également tous les fichiers dans les sous-répertoires
- **ls -a** affichera les fichiers cachés
- **ls -al** listera les fichiers et les répertoires avec des informations détaillées comme les autorisations, la taille, le propriétaire, etc.
- **ls mondossier** liste les fichiers dans mondossier

4. Commande cat

cat (abréviation de concatenate) est l'une des commandes [Linux](#) les plus fréquemment utilisées. Elle est utilisée pour lister le contenu d'un fichier sur le résultat standard (sdout). Pour exécuter cette commande, tapez **cat** suivi du nom du fichier et de son extension.

Par exemple : **cat fichier.txt**.

Voici d'autres façons d'utiliser la commande **cat** :

- **cat > nomDeFichier** crée un nouveau fichier
- **cat nomDeFichier1 nomDeFichier2>nomDeFichier3** joint deux fichiers (1 et 2) et enregistre le résultat de ces derniers dans un nouveau fichier (3)
- pour convertir un fichier en majuscules ou en minuscules, **cat nomDeFichier | tr a-z A-Z >resultat.txt**

5. Commande cp

Utilisez la commande **cp** pour copier les fichiers du répertoire actuel dans un autre répertoire. Par exemple, la commande **cp scenery.jpg /home/utilisateur/Photos** créera une copie de **scenery.jpg** (de votre répertoire actuel) dans le répertoire **Photos**.

6. Commande mkdir

Utilisez la commande **mkdir** pour créer un nouveau répertoire.

Si vous tapez **mkdir Music**, cela créera un répertoire appelé **Music**.

Il existe également des commandes **mkdir** supplémentaires :

- Pour générer un nouveau répertoire à l'intérieur d'un autre répertoire, utilisez cette commande de base de Linux **mkdir Music/Nouveau**
- utiliser l'option **p** (parents) pour créer un répertoire entre deux répertoires existants. Par exemple, **mkdir -p Musique/2020/Nouveau** créera le nouveau répertoire « 2020 ».

7. Commande rmdir

Si vous avez besoin de supprimer un répertoire, utilisez la commande **rmdir**. Cependant, **rmdir** ne vous permet de supprimer que les répertoires vides.

8. Commande rm

La commande **rm** est utilisée pour supprimer les répertoires et leur contenu. Si vous voulez seulement supprimer le répertoire – comme alternative à **rmdir** – utilisez **rm -r**.

Note : Soyez très prudent avec cette commande et vérifiez à nouveau dans quel répertoire vous vous trouvez. Cela effacera tout et il n'y aura pas d'annulation.

9. Commande touch

La commande **touch** vous permet de créer un nouveau fichier vierge via la ligne de commande [Linux](#). Par exemple, entrez **touch /home/username/Documents/Web.html**

pour créer un fichier HTML intitulé **Web** dans le répertoire **Documents**.

10. grep command

Une autre commande de base de [Linux](#) qui est sans aucun doute utile pour une utilisation quotidienne est **grep**. Elle vous permet de rechercher tout le texte d'un fichier donné.

Par exemple, **grep blue notepad.txt** recherchera le mot **blue** dans le fichier **notepad**. Les lignes qui contiennent le mot recherché s'afficheront entièrement.

11. Commande sudo

Un grand pouvoir implique de grandes responsabilités !

Abréviation de « **SuperUser Do** », cette commande vous permet d'effectuer des tâches qui nécessitent des autorisations administratives ou de root. Cependant, il n'est pas conseillé d'utiliser cette commande pour un usage quotidien car une erreur pourrait facilement se produire si vous avez fait quelque chose d'incorrect.

Le prompt de **sudo** est **#** qui signifie que vous êtes connecté en tant que **root** (administrateur du système) possédant tous les droits.

root@johndoe-desktop:~# apt-get upgrade

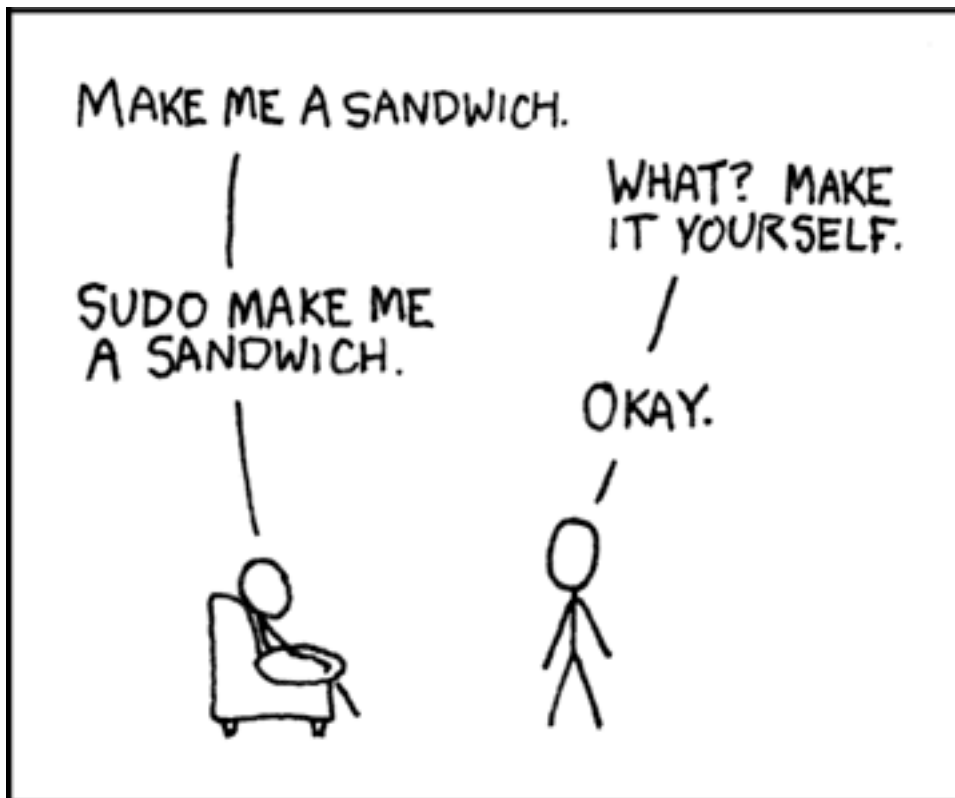
johndoe@johndoe-desktop~\$ sudo apt-get upgrade (root pour un instant)

sudo su devenir root et le rester

Pour quitter le « mode root », tapez **exit** (ou faites la combinaison **Ctrl + D**).

\$: signifie que vous êtes en train d'utiliser un compte utilisateur « normal »

```
sudo apt-get install
sudo apt-get update
sudo apt-get upgrade
sudo apt-get dist-upgrade
sudo apt-get check
```



12. Commande ping

Utilisez la commande **ping** pour vérifier votre état de connectivité à un serveur. Par exemple, en entrant simplement **ping google.com**, la commande vérifiera si vous êtes en mesure de vous connecter à Google et mesurera également le temps de réponse.

13. Commande uname

La commande **uname**, abréviation de Unix Name, imprimera des informations détaillées sur votre système Linux comme le nom de la machine, le système d'exploitation, le noyau, etc.

14. Commande history

Lorsque vous utilisez Linux depuis un certain temps, vous remarquerez rapidement que vous pouvez exécuter des centaines de commandes chaque jour. Ainsi, l'exécution de la commande **history** est particulièrement utile si vous voulez revoir les commandes que vous avez entrées auparavant.

Naviguer dans la recherche **history** : Flèche Haut, Flèche Bas

15. Commande man

Confus quant à la fonction de certaines commandes Linux ? Ne vous inquiétez pas, vous pouvez facilement apprendre à les utiliser directement depuis le shell de Linux en utilisant la commande **man**. Par exemple, en entrant la commande **man tail**, vous verrez les instructions manuelles de la commande tail.

16. Commande zip, unzip

Utilisez la commande **zip** pour compresser vos fichiers dans une archive zip, et utilisez la commande **unzip** pour extraire les fichiers zippés d'une archive zip.

17. Commande hostname

Si vous voulez connaître le nom de votre hôte/réseau, il vous suffit de taper **hostname**. En ajoutant un **-I** à la fin, vous obtiendrez l'adresse IP de votre réseau.

18. Commande useradd, userdel

Puisque Linux est un système multi-utilisateurs, cela signifie que plusieurs personnes peuvent interagir avec le même système en même temps. **useradd** est utilisé pour créer un nouvel utilisateur, tandis que **passwd** est l'ajout d'un mot de passe au compte de cet utilisateur. Pour ajouter une nouvelle personne nommée John, tapez **useradd John** et ensuite pour ajouter son mot de passe, tapez **passwd 123456789**.

La suppression d'un utilisateur est très similaire à l'ajout d'un nouvel utilisateur. Pour supprimer le compte d'un utilisateur, tapez, **userdel NomUtilisateur**

Conseils et astuces

Utilisez la commande **clear(Ctrl + L)** pour nettoyer le terminal s'il est encombré par trop de commandes Linux déjà effectuées.

Essayez le bouton **TAB** pour remplir automatiquement ce que vous tapez. Par exemple, si vous avez besoin de taper Documents, commencez à taper une commande (disons **cd Docu**, puis appuyez sur la touche **TAB**) et le terminal remplira le reste, ce qui vous donnera **cd Documents**.

Ctrl+C et **Ctrl+Z** sont utilisés pour arrêter toute commande qui fonctionne actuellement.

Ctrl+C arrêtera et terminera la commande, tandis que **Ctrl+Z** mettra simplement la commande en pause.

Si vous bloquez accidentellement votre terminal en utilisant **Ctrl+S**, il suffit de le débloquent avec **Ctrl+Q**.

Vous pouvez exécuter plusieurs commandes en une seule en utilisant « ; » pour les séparer. Par exemple, **Commande1; Commande2; Commande3**. Ou utilisez **&&** si vous souhaitez que la commande suivante ne soit exécutée que lorsque la première a réussi.

Ctrl + A : ramène le curseur au début de la commande. La touche Origine a le même effet (elle est située à côté de la touche Fin et représentée par une flèche pointant en haut à gauche).

Ctrl + E : ramène le curseur à la fin de la ligne de commandes. La touche Fin a le même effet.

Ctrl + U : supprime tout ce qui se trouve à gauche du curseur. Si celui-ci est situé à la fin de la ligne, cette dernière sera donc supprimée.

Ctrl + K : supprime tout ce qui se trouve à droite du curseur. S'il est situé au début de la ligne, celle-ci sera donc totalement supprimée.

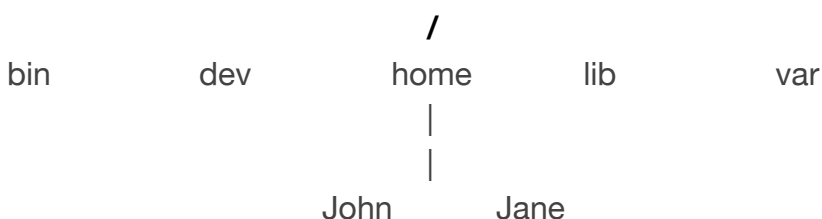
Ctrl + W : supprime le premier mot situé à gauche du curseur. Un « mot » est séparé par des espaces ; on s'en sert en général pour supprimer le paramètre situé à gauche du curseur.

Ctrl + R : recherche parmi les commandes précédentes.

passwd : changer le mot de passe

Organisation de Linux

Dossier racine : /



VS Code

Installer les plugins suivants :

French, ESLint, Live Server, Live Share, Emmet

Installer Node.js if not!

<https://css-tricks.com/gulp-for-beginners/>

Installer gulp (globalement -g) dans le terminal Linux

```
npm install -g gulp
```

```
npm install gulp gulp-sass
```

Terminal > Configure Tasks and click Create tasks.json file from template

Copier dans task.json

```
// Sass configuration
var gulp = require('gulp');
var sass = require('gulp-sass')(require('sass'));
var autoprefixer = require('gulp-autoprefixer');
```

```
gulp.task('sass', function(cb) {
  gulp
    .src('*.scss')
    .pipe(sass())
    .pipe(autoprefixer())

    .pipe(
      gulp.dest(function(f) {
        return f.base;
      })
    );
  cb();
});
```

```
gulp.task(
  'default',
  gulp.series('sass', function(cb) {
    gulp.watch('*.scss', gulp.series('sass'));
    cb();
  })
);
```

Modifier src destination

liveSassCompile.settings.formats

Settings --> Select "Live Sass Compiler Config --> add

Watch sass