TD1-1 : Premiers pas sous Linux
V2.4.1

Document en ligne : www.mickael-martin-nevot.com

1 Vocabulaire

- **Programme** (informatique) : séquence d'instructions exécutable par un ordinateur.
- **Processus** (informatique) : programme en cours d'exécution (« chargé » en mémoire vive).
- **Logiciel** : un ou plusieurs programme(s) accompagné(s) de données et de documentation.
- **Système d'exploitation** : ensemble de programmes et de logiciels (systèmes) contrôlant le matériel informatique et orchestrant les applications.
- **Application** (logiciel applicatif) : logiciel non système.
- **Système de fichiers** : mécanisme de stockage et d'accession des données sur « disque » (disque dur, clef USB, DVD-ROM, etc.).

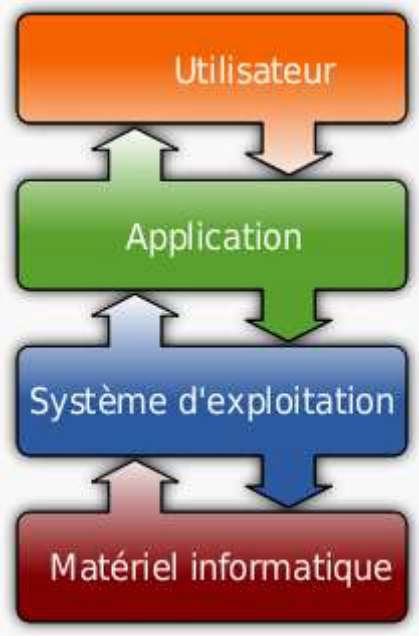

Figure 1 – Système d'exploitation et application

2 Le boot (amorce) : procédure de démarrage de l'ordinateur

Au démarrage de l'ordinateur, le *boot* (ou *bootstrap*) est la procédure d'amorçage permettant notamment d'exécuter le processus initial chargé d'exécuter lui-même le système d'exploitation (ou le *bootloader*).

Il est possible d'avoir plusieurs systèmes d'exploitation sur un ordinateur mais il est nécessaire de choisir celui que l'on utilisera à chaque démarrage. Un *bootloader* (ou chargeur d'amorçage) est un logiciel chargé de lancer un ou plusieurs système(s) d'exploitation (*multi-boot*).

Linux (ou GNU/Linux) est un système d'exploitation libre et gratuit de type UNIX (UNIX est à Mac OS et à Linux ce que MS-DOS est à Windows…), multitâche (exécution possible de plusieurs processus en même temps) et multi-utilisateur très utilisé dans le monde professionnel.

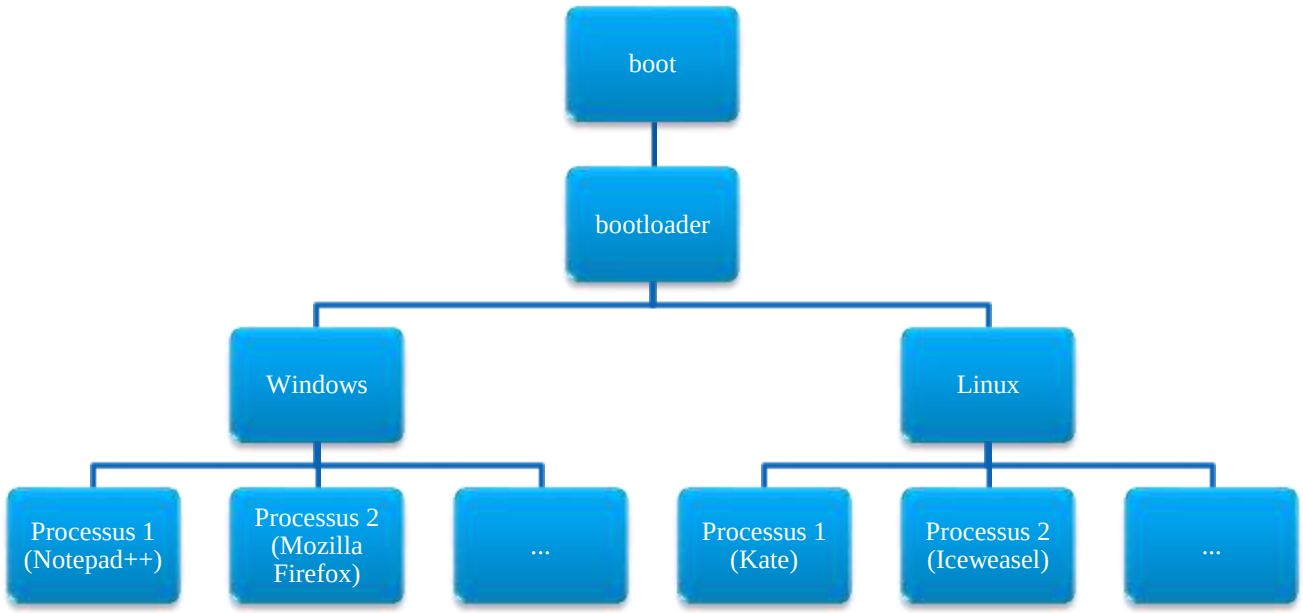

Figure 2 – Procédure de démarrage d'un ordinateur

Contrairement à Windows et Mac OS, Linux est disponible sous plusieurs distributions (ensembles cohérents de logiciels assemblés autour du noyau Linux : les logiciels système principaux).

De même, il existe plusieurs environnements graphiques (ou gestionnaires de bureau) différents sous Linux alors qu'il n'y en a qu'un seul sous Windows et Max OS.

Dans le cadre de cet enseignement, vous utilisez la distribution Debian et le gestionnaire de bureau GNOME. Il vous est bien sûr conseillé de découvrir par vous-même d'autres configurations.

3 Connexion sous Linux

Pour démarrer une session (de travail) à partir de l'invite de connexion, vous devez vous connecter au système en précisant votre nom d'utilisateur et votre mot de passe associé. Le bouton `option` vous permet de sélectionner le gestionnaire de bureau pour la session courante.

Choisissez donc GNOME, puis identifiez-vous.

4 Découverte du système de fichiers Linux via GNOME

Familiarisez-vous avec le gestionnaire de bureau GNOME (pour les utilisateurs d'autres systèmes d'exploitation : retrouvez vos marques et comparez).

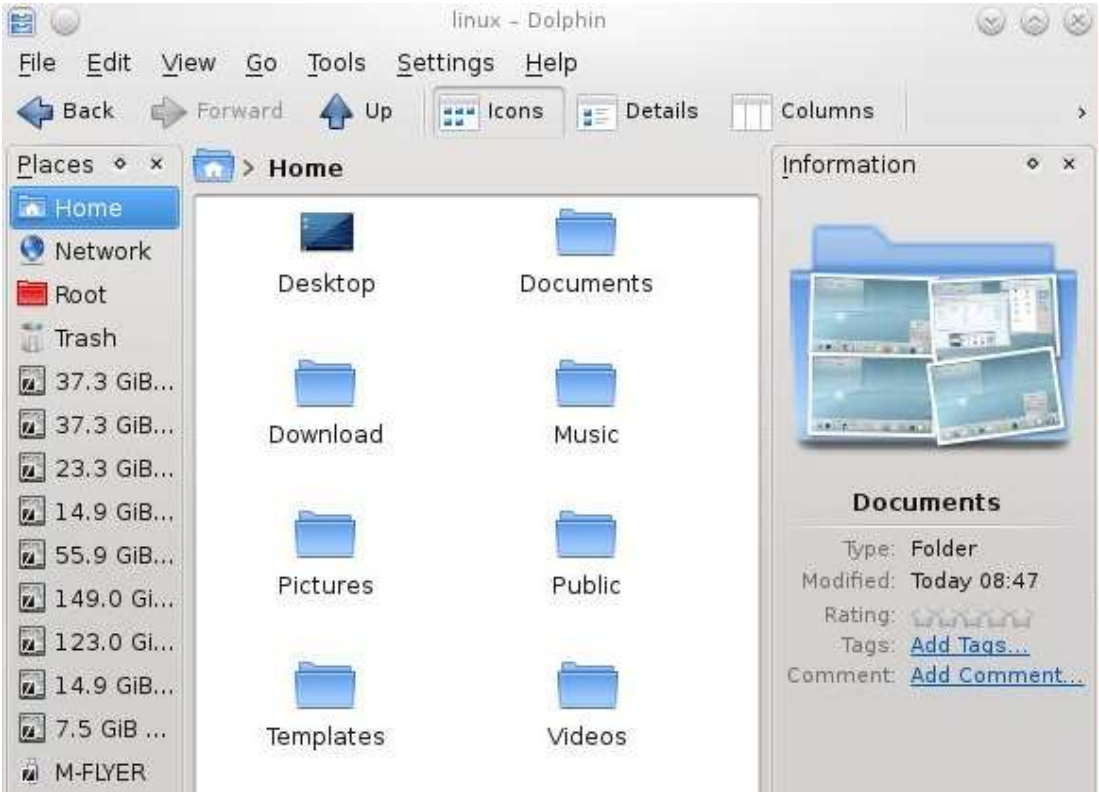

Figure 3 – Gestionnaire de fichiers Dolphin

En utilisant le gestionnaire de fichiers Dolphin :

- déterminez dans quel répertoire (aussi appelé dossier) est contenu votre répertoire principal (ou *home*) en utilisant l'onglet `aller` dans la barre de menu ;
- vérifiez quels répertoires et fichiers sont contenus dans votre répertoire *home* sans oublier d'afficher les fichiers cachés, c.-à-d. les fichiers commençant par un `.` sous Linux ;
- créez un fichier de test dans votre *home* ; vérifiez que cela a bien fonctionné et supprimez-le ;
- créez l'arborescence suivante à partir de votre *home* (reproduisez-la fidèlement) :

```
→ initiation-informatique
    → tp1
    → tp2
    → tp3
    → tp4
```

Pensez toujours, lors de vos différents TD et TP, à bien organiser vos fichiers afin de pouvoir les retrouver facilement : vous en aurez un grand nombre à l'issue de votre formation !

Enfin, sachez que le système de fichiers Linux est sensible à la casse : cela signifie qu'une chaîne de caractères change de sens selon que les lettres qui la composent sont en capitales (majuscules) ou en bas-de-casse (minuscules). Évidemment, il en va de même pour un point, une virgule, un tiret, une espace, etc. : tout est important sous Linux et dans l'immense majorité des domaines informatiques ; si deux chaînes de caractères ne sont pas exactement les mêmes, elles seront considérées comme réellement différentes ! Il faut donc prêter une attention toute particulière à la création de vos répertoires et à l'écriture de vos fichiers. De manière générale une grande rigueur en informatique est de mise pour éviter les erreurs.

Sans mention contraire, vous vous positionnez dans votre (sous-) répertoire `tp1` durant toute la suite de ce TD.

5 Premier contact avec l'interpréteur de commande

Les interpréteurs de lignes de commande (ou *shells*) tels que Bash (*bourne-again shell*) font partie des programmes les plus stables qui puissent tourner sur un système d'exploitation Linux. Leur avantage principal est qu'ils permettent de travailler plus rapidement qu'à partir d'un environnement graphique (et même bien plus rapidement une fois maitrisés). Vous allez apprendre à utiliser quelques commandes simples.

Lancez une console (ou terminal). Un *shell* Bash s'exécutera automatiquement dans la console :
`menu K → Applications → Système → Terminal`

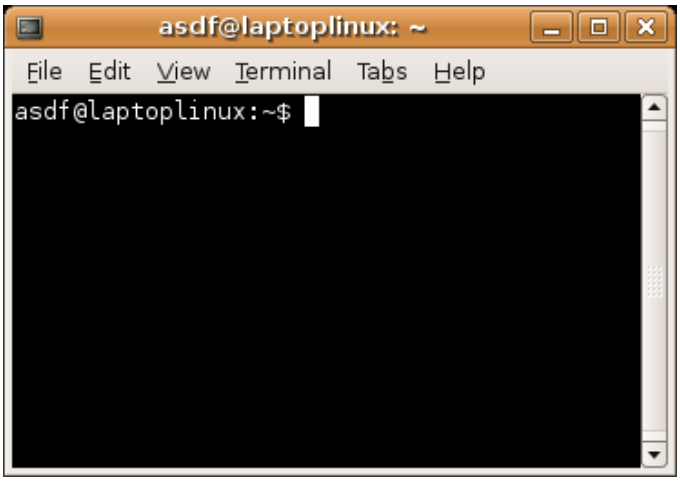

Figure 4 – Exemple de console

Au lancement du *shell*, vous pouvez voir une ligne terminant par `$` : c'est l'invite de commande (ou *prompt*). Celle-ci indique à l'utilisateur que le *shell* est en attente d'une commande. Si l'invite de commande n'apparaît pas, c'est que votre *shell* est déjà occupé à exécuter une autre commande : il est donc inutilisable jusqu'à ce que le *prompt* réapparaisse.

Pour utiliser (ou lancer) une commande, il faut employer le **nom de la commande** suivi éventuellement d'un ou plusieurs « mots » **séparés par des espaces**, et terminer en pressant la touche `Entrée` (qui demande au *prompt* l'exécution de la commande). Ces « mots » peuvent être des **options** (le plus souvent des lettres précédées du caractère −) ou des **arguments**.
Par exemple (`cmd` étant une commande, `opt1` et `opt2` des options et `par1` et `par2` des arguments) :

```
cmd -opt1opt2 par1 par2
```

La syntaxe d'utilisation des commandes peut légèrement varier d'un *shell* à un autre même si la majorité de celles vues dans le cadre de cet enseignement sont standards.

Voici quelques commandes de base :

- `man` : permet de visionner le manuel d'une commande mise en argument (par exemple : `man pwd`) ; certaines pages du manuel sont traduites en français, d'autres non (et sont donc en anglais) ; pressez la touche `q` pour quitter le manuel ; n'oubliez pas de consulter le manuel à chaque fois que cela est nécessaire, en particulier à chaque découverte d'une nouvelle commande ;
- `pwd` : affiche le nom complet du répertoire en cours : comparez le résultat obtenu avec celui de votre voisin(e) ;
- `whoami` : cherchez l'utilité de cette commande dans le manuel et testez ;
- `ls` : cherchez l'utilité de cette commande dans le manuel et testez les lignes de commande `ls -a`, `ls -l`, `ls -a -l`, `ls -l -a`, `ls -al`, `ls -la`, puis `ls -al --color=auto` (utilisation singulière d'une option) : comparez les résultats, puis vérifiez que vos déductions soient correctes en utilisant le gestionnaire de fichier de GNOME ;
- `cat`, `more`, `less` : cherchez l'utilité de ces commandes (de la même famille) dans le manuel et testez-les pour connaitre leurs différences ;
- `ps` : cherchez l'utilité de cette commande dans le manuel et testez la ligne de commande `ps -f -U login` (en remplaçant `login` par votre *login*, ou identifiant).

Une commande peut aussi être un nom de programme. Par exemple il suffit d'utiliser la commande `iceweasel` pour lancer Iceweasel (la version renommée de Mozilla Firefox).

6 Arborescence, chemin absolu et relatif

6.1 Arborescence des fichiers Linux

Sous Linux, toutes les données (y compris s'il y a plusieurs disques durs par exemple) sont accessibles en suivant un chemin depuis un emplacement appelé la racine (ou *root*) noté `/`. Cet emplacement porte bien son nom puisque l'on peut voir l'arborescence des fichiers comme un arbre (inversé) nous permettant de nous déplacer le long des branches pour atteindre les feuilles.

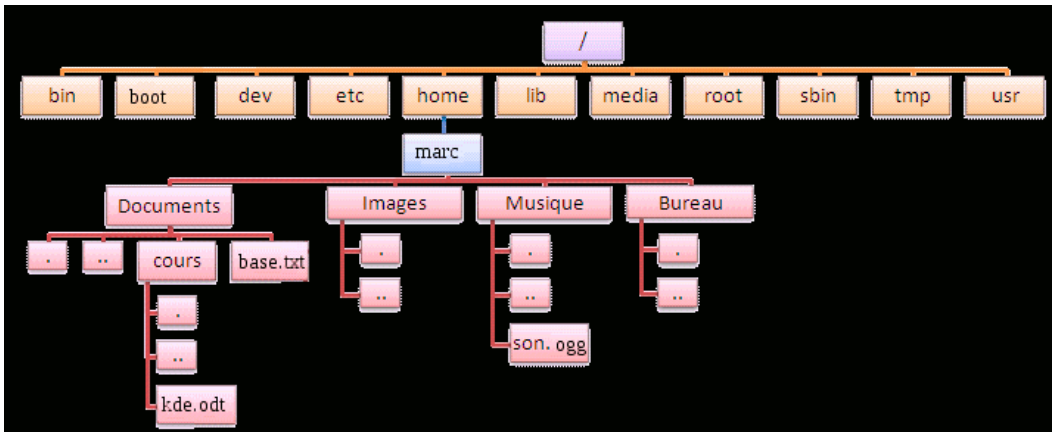

Figure 5 – Arborescence des fichiers Linux

6.2 Chemin absolu et chemin relatif

Un chemin est une chaîne de caractères décrivant la position (exacte) d'une ressource (fichier, répertoire, etc.). Il existe deux manières **équivalentes** de connaitre le chemin d'une ressource : de manière absolue ou relative.

6.2.1 Chemin absolu

Un chemin absolu est préfixé par la racine (un `/` sous Linux) et fait donc toujours référence à la racine de l'arborescence des fichiers. L'avantage d'un chemin absolu est qu'il est invariant pour une même ressource, quel que soit le répertoire courant.

6.2.2 Chemin relatif

Il n'y a pas de racine en préfixe pour un chemin relatif : c'est une référence à la position relative de la ressource dans l'arborescence des fichiers, c.-à-d. par rapport au répertoire courant. Souvent plus court qu'un chemin absolu, un chemin relatif varie cependant le plus souvent lorsque le répertoire courant change.

6.3 Exercice

La commande `cd` permet de se déplacer dans l'arborescence des fichiers, c.-à-d. entrer dans un répertoire ou sortir d'un répertoire.

Pour entrer dans un répertoire, il suffit de taper la commande `cd` suivie d'une espace et du nom du répertoire cible, par exemple : `cd initiation-informatique`.

Déterminez ce que font les lignes de commande `cd ..`, `cd ~`, `cd .` et `cd`.

Ouvrez une console puis :

- entrez dans le répertoire `initiation-informatique` ; vérifiez que cela a bien fonctionné à l'aide de la commande `pwd` ;
- sortez du répertoire `initiation-informatique` ; vérifiez que cela a bien fonctionné ;
- placez-vous à présent dans le répertoire `tp2` en utilisant une seule ligne de commande ; vérifiez que cela a bien fonctionné ;
- enfin, en seulement deux lignes de commande, positionnez-vous sur le répertoire *root* (`/`) et listez le contenu du répertoire `/bin` ; vérifiez que cela a bien fonctionné.

7 Éditeur de texte

Kate est un éditeur de texte (logiciel destiné à la création et à l'édition de fichiers textes, incontournable en programmation et dans de nombreuses tâches informatiques) puissant et populaire.

Il ne faut pas confondre éditeur de texte et traitement de texte de type Microsoft Word. L'éditeur se distingue par le fait qu'il est orienté lignes de code plutôt que paragraphes, et que les fichiers textes ne contiennent pas de mise en forme (taille et genre de la police, etc.). Le traitement de texte a un format de fichiers élaboré, contenant les informations de présentation.

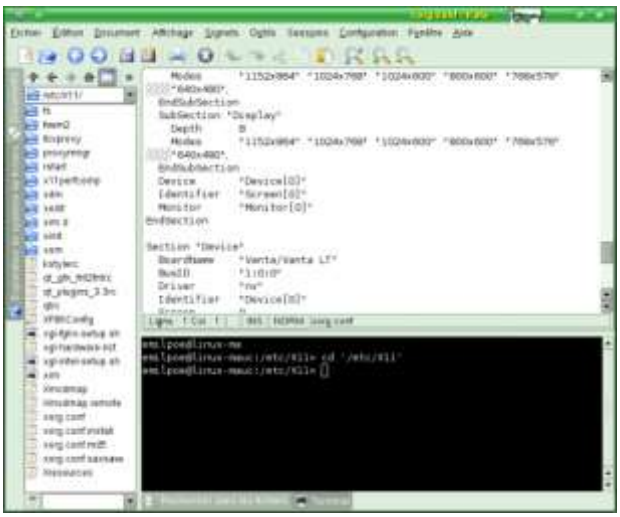

Figure 6 – Capture d'écran de kate

Comme le montre la capture d'écran, Kate est composée par défaut de l'arborescence de fichiers à gauche, d'une zone de saisie de texte à droite, d'une barre d'outils classique en haut et éventuellement d'un *shell* en bas à droite.

7.1 Fonctionnalités principales

Voici les fonctionnalités principales de Kate :

- coloration syntaxique (plus de cent formats pris en charge) ;
- pliage de code (*code folding*) ;
- édition possible de plusieurs fichiers simultanément ;
- intégration d'un *shell* dans l'interface ;
- capacité d'édition de fichiers de tailles importantes sans ralentissements majeurs ;
- possibilité d'utilisation de *plugins* (ou module d'extension).

7.2 Logiciels associés

Voici des versions dérivées de Kate :

- KWrite : version simplifiée de Kate ;
- KEdit : version élémentaire (très simplifiée) de Kate.

7.3 Exercice

Recherchez sur internet des informations sur le fonctionnement de Kate. Attention ! Il faut toujours vérifier que les informations trouvées sur le Web soient correctes. En particulier, déterminez quels raccourcis claviers correspondent aux actions suivantes :

- ouvrir un fichier ;
- sauvegarder sur « disque » ;
- quitter Kate ;
- copier/coller ;
- annuler/rétablir ;
- rechercher une séquence de lettres ;
- remplacer une séquence de lettres par une autre.

Familiarisez-vous le plus possible avec ce logiciel en utilisant les raccourcis clavier (cela vous fera gagner du temps) :

- commencez par afficher le nombre de lignes (très utile en programmation) ;
- créez un nouveau fichier `test.txt` avec Kate ;
- écrivez une phrase dans le fichier `test.txt` puis sauvegardez le fichier et quittez Kate ; vérifiez que vous avez bien sauvegardé le fichier `test.txt` (en exécutant la ligne de commande `cat test.txt` par exemple) ;
- exécutez de nouveau Kate : vous voyez votre phrase ; modifiez-la puis quittez Kate sans sauvegarder ;
- vérifiez que les changements n'aient pas été pris en compte en affichant le contenu du fichier `test.txt` (à l'aide de la commande `less` par exemple) ;
- sélectionnez une partie de votre texte, copiez-la, puis collez-la à la fin de votre texte ;
- recherchez un mot (autrement dit une chaîne de caractères) de votre texte puis l'occurrence suivante de cette chaîne si elle existe ; recommencez avec une autre chaîne de caractères ;
- créez un nouveau fichier texte avec Kate ; partagez la zone de saisie textuelle (verticalement puis horizontalement) de sorte à afficher simultanément vos deux fichiers textes (ce qui est parfois bien commode) ;
- en utilisant l'arborescence de fichiers de Kate, ouvrez plusieurs fichier textes ; naviguez ensuite de l'un à l'autre ;
- quittez Kate.

Il vous arrivera (notamment dans votre travail) de ne pas utiliser ce gestionnaire de bureau, et même aucun environnement graphique. Dans ce cas-là, vous serez amené à utiliser des éditeurs de texte en ligne de commande, tels que pico ou vi qui peuvent paraitre archaïques ou repoussants au premier abord mais qui sont pourtant populaires et puissants. Pour une raison de temps, ils ne seront pas vus dans le cadre de cet enseignement mais il vous est vivement conseillé d'apprendre leur maniement rudimentaire. De même, Geany et Emacs (ainsi que sa version évoluée XEmacs) sont des éditeurs de texte graphiques plus complexes que Kate mais cependant très utilisés.