

LICENCE 1  
MULTIMEDIA INFORMATION  
ET COMMUNICATION

SYSTEME D'EXPLOITATION

Chapitre 2: Les systèmes  
GNU/Linux

\*Enseignant concepteur : IVAN KALIA

## 2.1 Le projet GNU/Linux

Le GNU est le nom donné au système d'exploitation écrit par Richard M. Stallman qui débuta le projet en 1984 et publia la première version en 1989 sous licence GPL (General Public Licence, fondement éthique, juridique et politique du mouvement libre). Arrivé en 1991, le projet GNU rassemblait déjà plusieurs outils libres, mais il lui manquait un élément centrale: le noyau. C'est à ce moment là que Linus Torvalds (ancien étudiant finlandais) a décidé de contribuer au projet et développa un noyau en s'inspirant des systèmes UNIX qu'il a placé sous licence GPL. Ce noyau sera appelé Linux (concaténation de UNIX et Linus). Le système d'exploitation regroupant les outils GNU fonctionnant sur le noyau Linux s'appellera GNU/Linux (GNU sur Linux) [5].

L'objectif de ce chapitre est de permettre aux étudiants en MIC de se familiariser à l'utilisation et la personnalisation d'un système Linux.

## 2.2 Qu'est-ce qu'un logiciel libre ?

L'expression « Logiciel Libre » fait référence à la liberté d'utilisation et non pas aux prix. Pour comprendre le concept, vous devez penser à la « liberté d'expression », pas à « l'entrée libre ».

L'expression « Logiciel Libre » fait référence à la liberté pour les utilisateurs d'exécuter, de copier, de distribuer, d'étudier, de modifier et d'améliorer le logiciel.

Plus précisément, elle fait référence à quatre types de liberté pour l'utilisateur du logiciel :

- ✓ **Liberté 0:** La liberté d'exécuter le programme, pour tous les usages.
- ✓ **Liberté 1:** La liberté d'étudier le fonctionnement du programme, et de l'adapter à vos besoins.
- ✓ **Liberté 2:** La liberté de redistribuer des copies, donc d'aider votre voisin.
- ✓ **Liberté 3:** La liberté d'améliorer le programme et de publier vos améliorations, pour en faire profiter toute la communauté. Pour se faire, l'accès au code source est une condition requise. Un programme est un Logiciel Libre si les utilisateurs ont toutes ces libertés [5].

## 2.3 Concepts de base sous Linux

Linux est un système d'exploitation orienté fichiers contrairement au système MS-DOS1 (Windows Microsoft) ou OpenVms2 qui sont plutôt orientés disque. Sous Linux, il n'existe pas sous Linux au niveau utilisateur, la notion de disque physique. On ne voit qu'une seule arborescence dont différents points sont rattachés sur un système de fichiers. Celui-ci fait correspondre physiquement:

- ✓ une partition d'un disque physique
- ✓ un ensemble de disque physique (volume logique)
- ✓ un système de fichier sur une machine distante (service NFS par exemple) [3]. Linux respecte la casse. Les commandes sont en caractère minuscule et les variables systèmes ou variables d'environnement sont en majuscule. Par exemple, “ls” et “LS” ou “Ls” ou encore “IS” n'ont pas les mêmes significations. Ce sont quatre mots différents au niveau de l'interpréteur. Chaque commande Linux crée un processus. Lorsqu'une commande est exécutée au niveau de l'interpréteur de commandes, celui-ci crée un processus fils qui s'exécute dans l'environnement du processus père. Dès que l'exécution du processus fils est terminée, ce dernier rend la main au processus père.

### 2.3 Caractéristiques des systèmes Linux

Le rôle fondamentale d'un système est de:

- ✓ Gérer le matériel
- ✓ Masquer le matériel en permettant aux programmes d'interagir avec le matériel à travers des pilotes
- ✓ Partager les ressources entre les différents programmes en cours d'exécution (processus).
- ✓ Fournir une interface pour les programmes (un ensemble d'appels système)
- ✓ Fournir une interfaces bas niveau pour l'utilisateur (un ensemble de commande Shell)
- ✓ Eventuellement une interface utilisateur haut niveau par un environnement graphique (kde, gnome).. (voir figure ci-dessous)

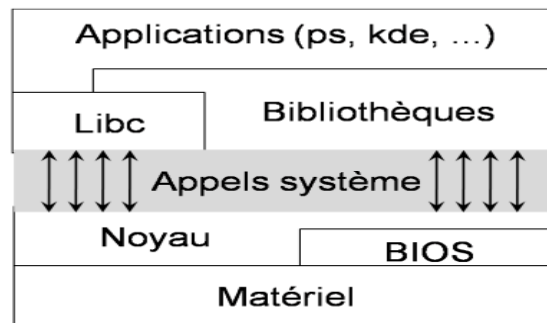
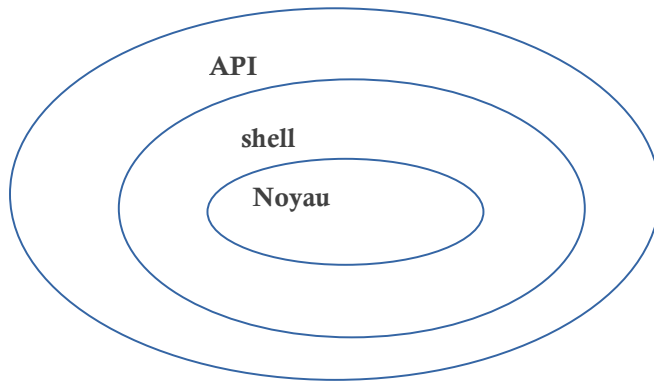


Figure 2.1: Schéma d'un système d'exploitation Linux [3]

Un système linux est constitué d'un noyau, d'un interpréteur appelé Shell et d'un ensemble de bibliothèques

organisés sous forme d'API (Application Programming Interface).



**Figure 2.2 : Architecture simplifiée d'un système Linux**

- ✓ Le noyau de système d'exploitation (abrégé noyau, ou kernel en anglais), est la partie fondamentale des systèmes d'exploitation (en général). Elle gère les ressources de l'ordinateur et permet aux différents composants matériels et logiciels de communiquer entre eux.
- ✓ Le Shell est un interpréteur de commande. Il met un environnement d'exécution à disposition de l'utilisateur lui permettant d'exécuter des commandes qui seront envoyées au noyau.
- ✓ Les API (Application Programming Interface) est un ensemble de bibliothèques permettant aux programmes (commandes) de s'exécuter convenablement.

Les caractéristiques essentielles d'un système Linux sont :

- ✓ Multitâche : un système Linux permet l'exécution de plusieurs processus simultanément à la fois
- ✓ Multiutilisateur : plusieurs utilisateurs peuvent se connecter simultanément sur le système à la fois
- ✓ système hiérarchisé : les fichiers sont organisés sous forme d'arborescence dont la racine est symbolisée par « / ».

## 2.4 Connexion déconnexion

### 2.4.1 Les types de comptes

Les systèmes UNIX/Linux comme les systèmes Windows (Microsoft) définissent généralement trois types de comptes:

- ✓ Le compte invité (guest) : c'est aussi un compte utilisateur, mais ce compte est défini par le système et

n'ayant pas assez de privilèges sur le système. Généralement sous Linux, lorsque vous ouvrez une session utilisateur sous le compte « guest ou invité », vous ne pouvez pas prendre les privilèges du super-utilisateur pour exécuter les tâches d'administration.

- ✓ Le compte utilisateur : c'est le compte défini par l'utilisateur lors de l'installation du système. C'est sous ce compte que l'utilisateur ouvre les sessions sur le système. Le compte utilisateur a aussi des droits réduits sur le système. Le prompt d'un simple utilisateur est symbolisé par "\$".
- ✓ Le compte super-utilisateur : Encore appelé compte "root" ou « administrateur » sous Windows, ce compte dispose de tous les droits sur le système et son prompt est symbolisé par "#".

### - 2.4.2 Connexion

Après l'installation du système, tout utilisateur doit s'authentifier pour ouvrir une session sur le système. C'est ce qu'on appelle « la phase d'identification ». Lors de la phase d'identification, tout utilisateur est appelé à fournir ces paramètres d'authentification (login et mot de passe). Une fois identifié, l'utilisateur est autorisé à ouvrir la session sur le système.

L'ouverture de session peut se faire à distance ou en local sur le même système. Généralement, l'accès au système Linux est caractérisé par deux modes :

#### ✓ En mode graphique

Lors d'une ouverture de session, l'utilisateur se trouve en face d'un premier écran d'identification. Cette identification consiste à authentifier l'utilisateur en vérifiant la cohérence de ses paramètres d'authentification basés sur un **login** et un **mot de passe**. Les utilisateurs sont souvent très familiers à l'interface graphique, car elle est facile à manipuler. Par conséquent, il n'est pas évident de régler certains conflits système par exemple la vérification de l'intégrité des systèmes de fichiers (des outils graphiques sont souvent mis à profit) en mode graphique, mais plus aisé pour un utilisateur qui maîtrise le système [2].

Comme tout système multiutilisateurs et multitâches, le terme opposé est système mono-utilisateur qui est utilisé lorsque l'on parle de systèmes d'exploitation utilisables par un seul utilisateur à la fois ou encore en référence à une licence de logiciel prévue pour un utilisateur et mono-programmé (un seul programme s'exécute) [3].

Linux associe à chaque utilisateur :

- ✓ un nom appelé "login" (équivalent au "Username" de Open VMS, ou au "logon name" de Windows—

NT),

- ✓ un mot de passe,
- ✓ un numéro d'utilisateur unique ou "UID" (équivalent à l'UID Open VMS ou "logon ID" de Windows-NT).

Par conséquent, la première chose que vous demandera le système sera votre "login" et votre mot de passe. Si les deux sont valides, Unix initialisera votre environnement de travail (voir figure ci-dessous).

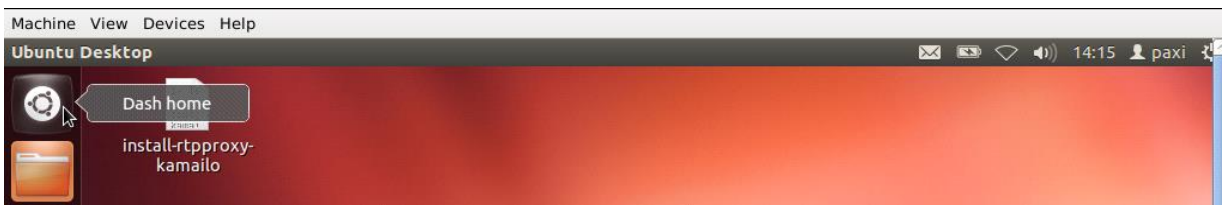


**Figure 2.3: Connexion utilisateur en mode graphique**

### 2.4.3 Travail en ligne de commandes

Un utilisateur peut également ouvrir une session utilisateur en mode texte (console). L'avantage de la ligne de commande permet de réaliser les tâches de maintenance, des tâches plus complexes. La maîtrise des systèmes Linux passe par une aisance dans l'exécution des tâches en ligne de commandes.

La figure ci-dessous montre une session utilisateur en mode texte.



Les systèmes Linux ont amélioré leur interface graphique pour faciliter les tâches à l'utilisateur final. Pour se connecter en mode console, l'utilisateur doit cliquer dans la barre de menu (équivalent menu démarrer de Windows) et taper terminal



Pour effectuer des tâches d'administration, l'utilisateur est obligé de prendre les privilèges du super utilisateur (root). Pour cela, il exécute la commande "sudo -i" (super user do) et l'option -i permet d'initialiser le mot de passe pour root (sous ubuntu, lors de l'installation, le mot de passe de root n'est pas défini).



### 2.4.4 Les interfaces graphiques

En mode graphique, le programme d'ouverture de session propose de choisir le bureau qui sera l'environnement graphique de l'utilisateur (GUI pour Graphical User Interface). Les bureaux permettent de gérer l'apparence des fenêtres, et d'une manière générale l'apparence et le comportement de l'environnement de travail de l'utilisateur (comme la barre des tâches, les icônes et les menus des programmes, mais aussi le gestionnaire de fichiers, le navigateur, les signets, les outils de configuration, etc.). Les bureaux constituent un ensemble de logiciels et de bibliothèques ("Library") qui sont utilisables par d'autres programmes.

- GNOME (GNU Network Object Model Environment)
- KDE (K Desktop environment)

✓ Le bureau permet d'organiser l'environnement personnel de l'utilisateur. Les interfaces graphiques peuvent être configurées et personnalisées à l'aide d'outils de configuration comme "Le Centre de Contrôle". Un répertoire caché à l'intérieur du répertoire personnel de l'utilisateur conserve les préférences et les différents paramètres de l'utilisateur.

✓ La suppression de ce répertoire permet de revenir après un redémarrage aux réglages par défaut. A l'intérieur des interfaces graphiques de GNOME et de KDE, il est possible de gérer plusieurs "bureaux"

distincts (numérotés de 1 à 4, et accessibles avec la combinaison de touche ALT + F1 à F4) afin de disposer d'un espace de travail plus important. Les fenêtres des programmes peuvent être ainsi réparties sur les différents bureaux afin d'améliorer leur lisibilité et leur accessibilité.

✓ L'utilitaire "kpager" permet de gérer l'affichage des différents bureaux. Les gestionnaires de fichiers (fm pour "file manager") sont des outils graphique de navigation à l'intérieur de l'arborescence du système de fichiers.

- kfm (pour KDE)
  - gmc (pour GNOME, Graphical Midnight Commander)
  - mc (pour Midnight Commander sous console en mode texte, saisir "mc -c" pour ouvrir deux volets)
- ✓ Le bureau de KDE dispose de nombreux outils:
- Le gestionnaire de fichier ("kfm")
  - La suite bureautique "K Office" [w2].

### 2.4.5 Les environnements de bureau

✓ Un Environnement de bureau est un outil vous permettant d'exploiter les caractéristiques graphiques de votre ordinateur. Il est constitué de plusieurs éléments :

- le *bureau*, en arrière-plan de votre écran, qui affiche une image d'arrière-plan et un ensemble d'icônes
  - le gestionnaire de fenêtre qui crée les cadres entourant les fenêtres, zones d'échange entre vous et votre ordinateur
  - les *barres de menus* et les *panneaux* associés qui vous permettent d'accéder à vos logiciels, d'afficher l'heure, de lister vos fenêtres ouvertes, etc. ;
  - le *gestionnaire de sessions* qui gère vos sessions ;
  - des outils graphiques qui vous permettent de contrôler votre ordinateur, de le configurer et de l'exploiter.
- ✓ Il est important de différencier l'environnement de bureau et le système d'exploitation : alors que le système d'exploitation gère les ressources de votre ordinateur (contrôle des disques durs, contrôle des processeurs, contrôle des cartes graphiques et audio, etc.), l'environnement de bureau permet d'interagir avec votre ordinateur par l'intermédiaire d'une interface graphique (déplacement à l'écran avec une souris, un clavier ou un pavé tactile).



- ✓ **Unity** est l'environnement par défaut d'ubuntu. Il s'agit d'une surcouche de GNOME développée par Canonical. Initialement destinée aux netbooks et aux écrans de petite taille, cette interface est désormais adaptée aux postes de travail. Elle s'appuie sur le gestionnaire de fenêtres compiz, ce qui lui permet de mettre en œuvre des effets graphiques intéressants.
- ✓ **Gnome Shell** est l'interface officielle conçue par le projet GNOME ( Network Object Model Environment) pour l'environnement de bureau GNOME 3. Son but est de repenser l'interaction entre l'utilisateur et la machine en proposant une interface épurée et moderne centrée autour des tâches courantes effectuées par l'utilisateur.
- ✓ **KDE** (K Desktop Environment) est l'environnement de Kubuntu. Il repose sur une bibliothèque graphique portable **Qt**. Généralement apprécié pour sa modularité et sa simplicité, il n'en offre pas moins de très nombreux effets visuels et possibilités de personnalisation. Certains lui reprochent parfois une certaine lourdeur (surtout graphique), alors que d'autres apprécient son ergonomie proche de l'interface de Windows.
- ✓ **Xfce** est l'environnement de Xubuntu. Généralement apprécié pour sa légèreté et sa simplicité, il n'en est pas moins complet et puissant. Cependant il apparaît à certains un peu spartiate.
- ✓ **LXDE** est un environnement encore plus léger que XFCE.
- ✓ **MATE** est un **fork** de GNOME 2 destiné à maintenir une interface proche des métaphores de bureaux traditionnelles.
- ✓ **Cinnamon** est une interface développée par l'équipe de Linux Mint pour GNOME 3 (donc construite sur la même base qu'Unity et GNOME Shell) qui vise à demeurer plus proche d'un bureau classique tout en offrant diverses mises à jour liées à l'évolution de GNOME. Très séduisant.
- ✓ **Enlightenment** autrefois un simple et beau gestionnaire de fenêtres, mais qui tend à devenir un environnement complet avec la version e17.
- ✓ NB : Toutes ces interfaces peuvent être installées sur un même système.

### - 2.3.3 Arborescence des fichiers sous Linux

- L'une des caractéristiques fondamentales des systèmes UNIX/Linux est la structure arborescente.

Les systèmes Windows aussi sont caractérisés par une structure arborescente. L'arborescence des systèmes de fichiers sous Linux est constituée d'une « *racine* » symbolisée par « / ». La racine est le sommet de la hiérarchie des répertoires. Il s'agit d'une arborescence logique, indépendante de l'implantation physique des divers sous-répertoires, qui peut s'étendre sur plusieurs partitions incluses sur un ou plusieurs disques, et même sur des disques réseaux. En dessous de la racine, se trouvent les répertoires prédéfinies créés lors de l'installation. Ces répertoires définissent l'environnement de travail de l'utilisateur.

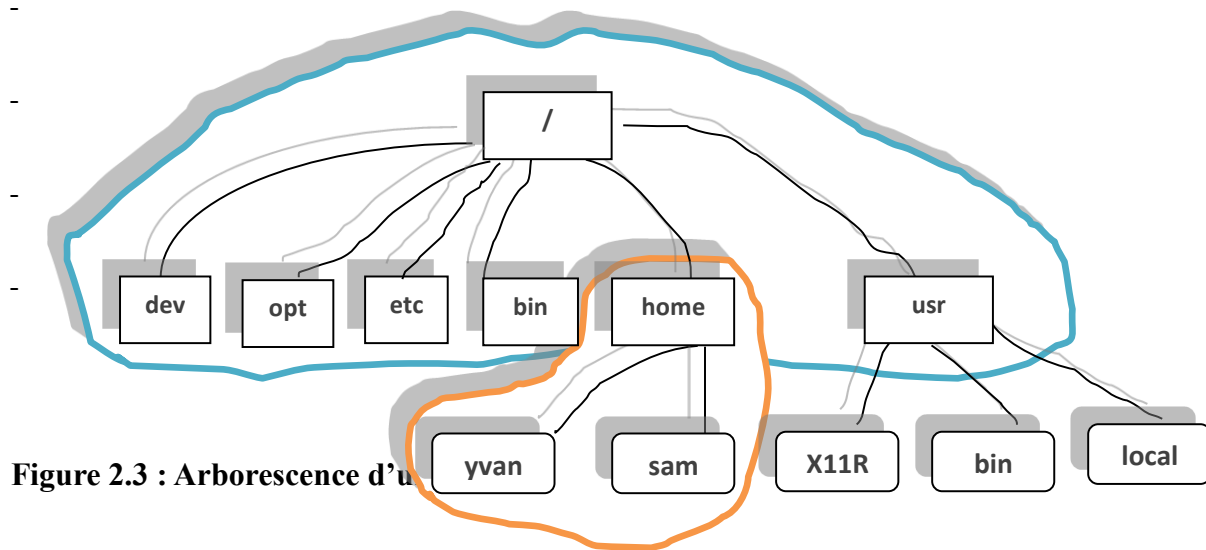


Figure 2.3 : Arborescence d'un système de fichiers

### 2.3.4 Les chemins d'accès

On distingue deux types de chemins d'accès aux fichiers :

- ✓ les chemins d'accès absolus,
- ✓ les chemins d'accès relatifs.

Les chemins d'accès absolus spécifient un nom de fichier ou de répertoire à partir de la racine.

Les chemins d'accès absolus commencent donc par "/" qui est le nom de la racine ou du root directory. Les chemins d'accès relatifs spécifient un nom de fichier ou de répertoire à partir du répertoire courant, appelés aussi working directory. Par conséquent, ces chemins ne commencent pas par "/" (exemple : dir/essai) [3].

- En résumé:
  - ✓ Chemins d'accès absolus : Commencent par "/", représentent la position relative par rapport au root directory.
  - Exemple :

**Exemple: `cd /home/abriel/Documents`**

- ✓ Chemins d'accès relatifs : Ne commencent pas par "/": Représentent la position relative par rapport au répertoire courant.

- Exemple :

```
abriel@server:~ /home/abriel $  
abriel@server:~$ pwd  
/home/abriel  
  
abriel@server:~ /home/abriel $ cd Telechargements
```

### 2.4 Les commandes de base

Une commande est constituée d'un nom suivi d'arguments. Le séparateur entre chaque mot d'une commande peut être un ou plusieurs espaces ou tabulations. Une commande peut s'exécuter sans option ou sans argument. Par conséquent, elle peut admettre plusieurs arguments (voir syntaxe).

Syntaxe :

**`cmd -[options] Arg1 Arg2 ....ArgN`**

Exemple:

```
abriel@server:~ /home/abriel $ mkdir -p cours1/cours2 cours3/cours4
```

### 2.4.1 Le Bash

BASH (Bourn Again Shel), est l'interpréteur par défaut des systèmes Linux. C'est le Shell par défaut, il permet d'exécuter des instructions que vous saisissez au clavier via le Shell ou au sein d'un script et vous en retourne les résultats. Nous ne détaillons pas les interpréteurs de commande dans ce chapitre.

Le Shell attend des entrées au clavier sur une ligne appelée l'invite de commande ou prompt à partir de l'entrée standard. Un curseur clignotant indique la position actuelle de votre saisie.

```
[abriel@serveur:/home/LPI]$
```

- ✓ *abriel*: c'est le nom de connexion ou le login
- ✓ *serveur*: le nom de la machine
- ✓ */home/LPI*: le répertoire de travail ou répertoire courant
- ✓ *\$*: le caractère de terminaison utilisateur simple: \$ ou >
- ✓ administrateur (root): #

### - 2.4.2 Manipulation de quelques commandes usuelles

Commandes	Description
echo	Affiche le résultat des données saisies en entrée standard
env	Affiche uniquement les variables d'environnement
set	Affiche les variables
sate	Affiche la date du système
cal	Cal affiche le calendrier du mois en cours. Des options associées permettent d'afficher le calendrier de l'année en cours

exec	Permet d'exécuter une commande sans lancer un processus
export	Permet d'exporter une variable du Shell dans les variables d'environnement
pwd	affiche en sortie standard le répertoire (chemin complet) courant
ls	Liste le contenu d'un répertoire en affichant le nombre de fichiers et répertoires
unset	Permet de supprimer une variable créée
man	Affiche le manuel de commande
uptime	Affiche la charge du système, temps pendant de démarrage, temps d'activité du système
w	Affiche le nombre d'utilisateur connecté
whoami	Affiche l'identité de l'utilisateur connecté
history	Affiche l'historique de toutes les commandes exécutées dans le Shell
diff	Concatène deux fichier et affiche en sortie standard leur différence
find	Utilisée pour rechercher un fichier dans l'arborescence du système
locate	Réalise les mêmes opérations que find mais utilise une base de données indexée
more	Affiche le contenu d'un fichier page par page
less	Même fonctionnalité que more
cp	Sert à copier des fichiers
mv	Permet de renommer ou de déplacer un fichier
type	Permet de connaître la nature d'une commande. Elle indique le fichier de la commande

id	Affiche les paramètres de l'utilisateur (UID, GID, UIDNumber, GIDNumber, groupe primaire, groupe secondaire s'il y'en a)
exit	Permet de se déconnecter du Shell
cmp	Permet de comparer deux fichiers
file	Permet de déterminer le type de fichier
cat	Concatène le contenu d'un fichier et l'affiche en sortie standard
ln (link)	Permet de créer un lien entre deux fichiers
join	Permet de joindre deux fichiers
paste	Concatène les champs de deux fichiers différents dans un même champ
head	Affiche par défaut les 10 premières lignes d'un fichier
df	Affiche l'espace disque disponible
du	Affiche l'espace occupé par un répertoire courant
nl	Numérote les lignes dans un fichier
fmt	Permet de formater un fichier
od	Convertit un fichier en octale
dd	Permet d'effectuer une copie bloc à bloc

✓ Connexion sous le compte root

Pour passer du simple utilisateur au superutilisateur, exécuter la commande ***sudo*** (super user do) ou ***su*** (***super user***).

```
abriel@server:~$ su -
Password:
root@server:~#
```

Pour se deconnecter sous le compte de root ou d'un autre compte, exécuter la commande ***exit***.

```
root@server:~# exit
logout
abriel@server:~$
```

### ✓ Affichage du contenu de répertoires avec ***ls*** (list)

```
abriel@server:~$ ls -l Documents/
total 12
drwxrwxr-x 5 abriel abriel 4096 sept. 30 11:18 cours11
drwxrwxr-x 3 abriel abriel 4096 sept. 30 10:57 doc4
drwxrwxr-x 2 abriel abriel 4096 sept. 30 11:17 doc5
-rw-rw-r-- 1 abriel abriel    0 sept. 27 11:36 fic2
```

### ✓ Affichage du répertoire courant avec ***pwd***

```
abriel@server:~$ pwd
/home/abriel
```

### ✓ déplacement dans l'arborescence avec ***cd***

- ***cd*** (chang directory): permet de changer de repertoire ou se déplacer dans une arborescence .

```
abriel@server:~$ cd /home/abriel/
abriel@server:~$ pwd
/home/abriel
```

- ***cd ..***: permet de quitter le répertoire courant

```
abriel@server:~/Documents$ cd ..
```

```
abriel@server:~$ pwd  
/home/abriel
```

### ✓ affichage identité utilisateur

- **whoami**: permet d'afficher l'identité de l'utilisateur connecté (le nom de connexion de l'utilisateur ou son username)

```
abriel@server:~$ whoami  
abriel
```

- **who** (qui?): affiche les noms des utilisateurs connectés, la liste des terminaux et l'heure à laquelle ils sont connectés

```
abriel@server:~$ who  
abriel    tty7          2016-09-30 10:45  
abriel    pts/1          2016-09-30 10:46 (:0)
```

- **date**: la commande affiche la date et l'heure du système. la sortie de cette commande affiche les paramètres suivants:
  - ✓ vendredi: le jour de la semaine
  - ✓ 30: le nombre de jour du mois
  - ✓ septembre: le mois
  - ✓ 2016: l'année
  - ✓ 11:42:19: l'heure de la journée
  - ✓ (UTC+0000): le fuseau horaire

```
abriel@server:~$ date  
vendredi 30 septembre 2016, 11:42:19 (UTC+0000)
```



- ✓ **cal**: affiche le calendrier du mois en cours

```
abriel@server:~$ cal
      Octobre 2016
di lu ma me je ve sa
                1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 31
```

- ✓ **env**: affiche les variables d'environnement

```
/sbin:/bin:/usr/games:/bin
  DESKTOP_SESSION=ubuntu-2d
  PWD=/home/abriel
  JAVA_HOME=/usr/lib/jvm/jdk
  GNOME_KEYRING_PID=2164
  LANG=fr_FR.UTF-8
```

- ✓ **echo**: elle permet d'afficher en sortie standard

```
abriel@server:~$ echo "Hello word students MIC"
Hello word students MIC
```

- ✓ **ps**: la commande ps (process) permet d'afficher la liste des processus lancés. cette commande s'exécute avec les options suivantes :
  - -a (--all): liste tous les processus lancés

- -u (--user): les processus lancés par l'utilisateur
- -x: pour désigner le terminal

```
abriel@server $ ps aux
```

✓ **kill**: la commande **kill** permet de mettre fin à l'exécution d'un processus

```
abriel@server $ kill -9 2020 (2020 est le PID du processus)
```

Voici quelques raccourcis claviers qui permettent de gérer les processus

- Ctrl+C: pour arrêter un processus lancé en front
- Ctrl+Z: pour arrêter l'exécution d'un processus
- bg % n° job: pour renvoyer un processus en tâche de fond

✓ Création de fichiers et répertoires

- Création de fichiers

il y'a deux manières de créer les fichiers. On peut créer un fichier vide en utilisant la commande **"touch"**.

```
Syntaxe: touch fic1 fic2 fic3.....ficN
```

```
exemple:
```

```
abriel@server:~/Documents$ touch fic1 fic2 fic3 fic4
```

```
abriel@server:~/Documents$ ls
```

```
fic1 fic2 fic3 fic4
```

- création de répertoires

- ✓ **mkdir** (make directory): la commande **mkdir** permet de créer un dossier ou un répertoire vide. on peut créer plusieurs répertoires a partir d'une seule commande

```
abriel@server:~/Documents$ mkdir doc1 doc2 doc3 doc4
abriel@server:~/Documents$ ls
    doc1  doc2  doc3  doc4
```

- ✓ Déplacement et renommage de fichiers et répertoires

- Déplacement de fichiers et repertoire

- **mv** (moov): la commande mv permet de déplacer les fichiers et répertoires.

syntaxe:

```
mv <source> <dest>
```

```
abriel@server:~/Documents/doc3$ mv ../fic1 ../fic3 .
abriel@server:~/Documents/doc3$ ls
    doc1  fic1  fic3
```

- ✓ **mv** permet aussi de réaliser le renommage des fichiers

syntaxe:

```
mv <old_name> <new_name>
```

```
abriel@server:~/Documents$ mv doc3 cours11
```

```
abriel@server:~/Documents$ ls
cours11  doc4  fic2  fic4
```

✓ copie de fichiers

- **cp** (copy): la commande cp permet de copier un fichier d'une source à une destination.

syntaxe:

```
cp <source> <dest>
```

```
abriel@server:~/Documents$ cp fic4 /home/abriel/Bureau/
abriel@server:~/Documents$ ls /home/abriel/Bureau/
fic4
```

✓ copie de répertoires

La commande cp permet aussi de copier un repertoire d'une source a une destination.

syntaxe:

```
cp <rep_source> <rep_dest>
```

```
abriel@server:~/Documents$ cp fic4 doc5/
abriel@server:~/Documents$ ls
cours11  doc4  doc5  fic2  fic4

abriel@server:~/Documents$ cp -R doc5 cours11/
abriel@server:~/Documents$ ls
cours11  doc4  doc5  fic2  fic4
```

```
abriel@server:~/Documents$ cd cours11/  
abriel@server:~/Documents/cours11$ ls  
doc1 doc4 doc5 fic1
```

### 2.5 Installation et gestions de fichiers

Il existe plusieurs types d'installation de paquets sous Linux. Le choix du type d'installation dépend de l'utilisateur en fonction du logiciel à disposition. Les applications peuvent être installées de deux manières : la compilation des paquetages sources ou l'installation à partir des dépôts en utilisant les gestionnaires de paquets.

#### 2.5.1 Installation à partir des dépôts

Les distributions Linux utilisent des gestionnaires de paquets pour installer, désinstaller ou mettre à jour les applications. Ces applications sont le plus souvent précompilées et ne demandent plus de compilation. L'installation des logiciels ou applications sous Linux peut se faire en mode graphique ou à partir de la ligne de commande. Cependant, l'installation en ligne de commande est mieux, car en cas d'échec en mode graphique vous ne saurez pas vraiment si la source du paquet existe ou vous n'aurez pas assez d'informations sur le paquet. L'installation en ligne de commande est le moyen le plus efficace. Il existe deux grands gestionnaires de paquets : RPM et DPKG

#### 2.5.1 Gestionnaire de paquet RedHat RPM

RPM (**RedHad Package Manager**) est le gestionnaire de paquets utilisé sur les distributions RedHad et dérivées. La gestion des paquetages est réalisée par la commande « rpm ». Les paquetages pré-compilés gérés par RPM sont d'extension *.rpm*. Les programmes en format *rpm* portent la structure suivante :

*nom\_programme-version-architecture.rpm*

Exemple d'installation d'un paquetage au format rpm

```
#rpm -ivh logiciel-1.1-i686.rpm
```

Options courantes de la commande rpm

- - i (ou - -install) : installe un paquetage
- - U (ou - -update) : met à jour un paquetage déjà installé
- - e (ou - -erase) : désinstalle un paquetage
- - q (ou - -query) : affiche les informations sur un paquetage installé
- - V (ou - -version) : vérifie un paquetage installé
- - v (ou - -verbose) : affiche le commentaire

### 2.5.2 Gestionnaire de paquet DPKG

**DPKG** (Debian Package Manager) est l'outil de gestion de paquets sur les distributions Debian et dérivées. Il permet d'installer, de configurer, mettre à jour et de désinstaller les paquetages Debian. Les paquetages pré-compilés gérés par DPKG sont d'extension `.deb`. La commande « `dpkg` » permet de gérer les paquetages au format `.deb`.

*`nom_logiciel-version-architecture.deb`*

Exemple d'installation d'un paquetage sous Debian

```
# dpkg -i logiciel-1.2.1-i386.deb
```

Options courantes de la commande `dpkg`

- -i (ou - -install) : installe un paquetage
- -r (ou - -remove) : désinstalle l'application
- -L : liste les fichiers du paquetage
- `dpkg-reconfigure` : permet de reconfigurer un paquetage déjà installé

NB : Il existe d'autres options de la commande `dpkg`

Les options contrôlant les actions de la commande `dpkg` sont définies dans le fichier « `/etc/dpkg/dpkg.cfg` ». les fichiers contenant les informations de base de données de paquetages se trouvent dans le fichier « `/var/lib/dpkg` ».

### 3.1.3 APT (Advanced Package Tools)

Les systèmes Debian ont développé un autre outil d'installation de paquets appelé « APT, Advanced Package Tools ». Cet outil s'appuie sur dpkg pour installer les paquets. Il s'appuie sur des liens définis dans le fichier `/etc/apt/sources.list`. Les utilisateurs expérimentés peuvent ajouter des dépôts complémentaires dans ce fichier si le système n'arrive pas retrouver le paquet lorsque vous lancez une requête d'installation. Pour explorer les différentes options de la commande apt, il est important de consulter le manuel.

DPKG es doté d'un outil de gestion avancé APT (Advanced Package Tools). Cet outil utilise la commande “dpkg” pour installer les paquets. La commande *apt* permet d'installer les logiciels et applications à partir des dépôts. Les liens vers ces dépôts sont définis dans le fichier “*/etc/apt/sources.list*” [2].

Les options courantes de la commande apt sont :

- ✓ apt-get install <logiciel> : install le paquetage logiciel
- ✓ apt-get remove <logiciel> : désinstalle le paquetage logiciel
- ✓ apt-get --purge remove <logiciel> : désinstalle le paquetage logiciel ainsi que les fichiers de configuration.

### 2.5.3 Compilation des paquets

Hormis, les paquets précompilés ou ceux installés directement à partir des dépôts (repository), il y'a des paquets archivés qui requiert une compilation avant d'être installés. Ces paquets sont généralement archivés par l'outil “*tar (tape archive)*” et compressés à l'aide des outils de compression tels que : gz, gzip, bz ou bzip2. Il faut procéder d'abord par un désarchivage et une décompression avant de passer à la compilation. Le programme tar est disponible par défaut sous Ubuntu. Il fait partie de l'installation minimale. Pour tous les formats à base de Tar, vous verrez que les options de tar sont les mêmes [w2] :

- **c** : crée l'archive
- **x** : extrait l'archive
- **f** : utilise le fichier donné en paramètre
- **v** : active le mode « verbeux » (bavard, affiche ce qu'il fait).

Puis selon la compression souhaitée :

- **z** : ajoute la compression Gzip.

- **j** : ajoute la compression Bzip2.
- **J** : ajoute la compression Lzma.

### ✓ Archivage avec tar

```
abriel@server:~/Documents$ tar cvf doc5.tar doc5/
```

Puis, on va compresser avec gzip et bzip2

```
abriel@server:~/Documents$ tar zcvf doc5.tar.gz doc5/
```

```
abriel@server:~/Documents$ tar jcvf doc5.tar.bz2 doc5/
```

### ✓ Décharchivage avec tar

```
abriel@server:~/Documents$ tar xzvf doc5.tar.gz
```

### ✓ Compilation

Voici les trois étapes de la compilation après avoir désarchiver le paquet. Généralement dans le dossier du paquet obtenu après décharchivage et décompression, il y'a un fichier nommé texte nommé README.txt ou INSTALL, qui décrit les procédures à suivre s'il l'installation du paquet nécessite des prérequis.

- `./configure` : regarde la configuration matérielle ainsi que les dépendances nécessaires à l'installation du logiciel
- `make` : compile le logiciel
- `make install` : installe le logiciel



-

### - 2.6 Gestions des fichiers et répertoires

#### - 2.6.1 Droits et permissions sur les fichiers et répertoires

Linux est un système multitâche et multiutilisateur. Il est donc important que l'exploitation du système par les utilisateurs soit soumise à un système de contrôlant les opérations que chaque utilisateur peut effectuer sur les ressources. Sous Linux, tout est traduit sous forme de fichiers de la configuration matérielle aux périphériques physiques.

Les droits d'accès et les permissions sur les fichiers (ressources) sont essentiels à la sécurité du système. Chaque ressource est représentée par un fichier et dispose de permissions spécifiques pour trois catégories d'utilisateurs.

- ✓ Le propriétaire de la ressource, symbolisé par la lettre « *u* » comme « *user* »
- ✓ Le groupe propriétaire de la ressource, symbolisé par la lettre « *g* » comme « *group* »
- ✓ Les autres, symbolisés par « *o* » comme « *other* ».

Chaque ressource dispose des droits de :

- ✓ Lecture symbolisé par la lettre « *r* (*read*) »
- ✓ Ecriture, symbolisé par la lettre « *w* (*write*) »
- ✓ Exécution, symbolisé par la lettre « *x* (*eXecution*) »
- ✓ Eventuellement des droits spéciaux

#### - 2.6.2 Visualisation des droits sur les fichiers

La commande « *ls* (*list*) » appliquée à l'option « *-l* » permettant de lister en long le contenu d'un répertoire en affichant les attributs du fichier

Exemple :

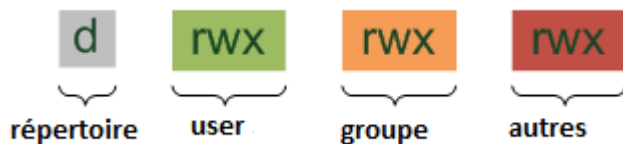
Affichage des attributs de fichier avec la commande « *ls -l* »

```
bash-4.1$ ls -l
total 1087
```

```
-rw-rw-r-- 1 yvan yvan 103 Jun 19 2014 blabla
-rw-rw-r-- 1 yvan yvan 74 Nov 30 2014 citations
-rw-r--r-- 1 root root 147096 May 12 2014 CMS - Wordpress.odp
lrwxrwxrwx 1 yvan yvan 7 Sep 15 14:33 coursinfo -> coursse
-rw-r--r-- 1 root root 29 Mar 2 2015 coursse
drwxr-xr-x. 35 yvan yvan 4096 Sep 8 17:32 Desktop
drwxr-xr-x. 36 yvan yvan 12288 Jul 28 13:51 Documents
```

Pour une ressource donnée, chaque catégorie d'utilisateur dispose de trois droits :

- ✓ Le droit de lecture
- ✓ Le droit d'écriture
- ✓ Le droit d'exécution



Chacun de ces droits en notation caractère dispose d'une valeur octale

Notation caractère	Notation octale
R	4
W	2
X	1
Rwx	7
r-x	5
r--	4

---	0
-----	---

Lorsqu'il s'agit d'un fichier, les droits sont faciles à interpréter :

- ✓ L'accès en lecture permet d'en consulter le contenu mais aussi de le copier.
- ✓ L'accès en écriture de le modifier.
- ✓ Et l'accès en exécution permet de tenter de l'exécuter (ce qui ne fonctionnera que s'il s'agit d'un programme).

Pour un répertoire, le traitement est différent

- ✓ L'accès en lecture donne le droit de consulter la liste de son contenu.
- ✓ L'accès en écriture celui d'y créer ou supprimer des fichiers.
- ✓ Et l'accès en exécution de le traverser (et notamment d'en faire le répertoire courant avec la commande "cd")

Voici les trois commandes de manipulation de fichiers et répertoires

La commande « *chmod* » permet de positionner les droits sur un fichier et répertoire.

Syntaxe :

*chmod* *-[option]* *<droit>* *<ressource>*

- chmod: options et parameters	
- Droit	- Préciser le droit qui doit être positionné en notation octale ou caractère
- Option	- Facultatif
- Ressource	- Désigne le fichier ou le répertoire pour lequel les droits vont être positionnés

Les droits peuvent être attribués en notation caractère ou octale. S'il s'agit d'ajout, on utilisera le signe (+) et le

signe (-) pour un retrait.

Exemple1 :

Positionnons les droits de lecture, écriture sur le répertoire « stage »

```
$ chmod u=rw,g=rw,o=rw
```

Exemple 2:

Retirons le droit de lecture au groupe propriétaire du dossier « stage » et attribuons à tous les droits d'exécution

```
# chmod u=r+w+x,g-r+w+x,o=r+w+x fichier
```

La commande « *chown (change owner)* » permet de changer l'appartenance d'un fichier

Syntaxe :

```
chown - [option] propriétaire :group_propriétaire <ressource>
```

- chown: options et parameters	
- Option	- Facultatif
- Ressource	- Fichier ou répertoire cible
- Propriétaire	- Designe le nom du nouveau propriétaire à donner à la ressource
- Group_propriétaire	- Designe le nom du nouveau group propriétaire de la ressource

Exemple 3 :

Changement de la propriété du fichier « stage »

```
# chown -R yvan :root stage
```

La commande « *chgrp (change group)* » s'exécute de la même manière que « *chown* ».

Exemple : Affectation du dossier « stage » au groupe « info »

```
# chgrp info stage
```

### 2.6.3 Présentations de quelques plateformes multimédias

- ✓ **Flumotion** est un projet de serveur de streaming vidéo open-source distribué sous licence GPL. Développé en Python et basé sur les frameworks Twisted et Gstreamer, il permet de diffuser sur un réseau des vidéos venant de sources lives (caméra, tv...) ou stockées dans des fichiers (on parle alors de VoD pour Video à la demande) en proposant une interface utilisateur de type Web [w3].
- ✓ **XBMC** est depuis plusieurs années porté sous /Linux, Windows et MacOS. Il est très complet, personnalisable avec de nombreux *skins* (thèmes d'apparence) et très abouti. Cette application supporte entre autres :
  - Une gestion de tous ses composants (applications , stream tv, thèmes, etc ...) comme des dépôts ou des appstores : vous pouvez par exemple ajouter des thèmes directement à partir d'XBMC, selon vos dépôts ; que vous pouvez également ajouter.
  - Les skins dont certains sont des institutions : Aeon et ses dérivées
  - Les webtv françaises et du monde entier, et les tv des box de fournisseur Internet
  - Il existe des composants pour le web, pour "réseaux sociaux", navigateur internet, etc ...
  - Des add-ons pour des applications Ubuntu, et les jeux, et les émulateurs etc ...
  - L'UPNP (Une technologie permettant entre autres de fournir et d'acquérir des médias sur un réseau local)
  - Le « Remote management » ( et la domotique), soit par Web interface, soit par diverses

applications (comme sur l'iPhone), soit par télécommande

- La réception de la Télévision/Radio Numérique grâce à une carte TV Pour cela, pensez à installer la version PVR.
- ✓ XBMC utilise l'accélération graphique afin de fournir de jolis effets de transition et de meilleures performances) De plus, XBMC peut également être exécuté comme gestionnaire de Bureau [w4].
- ✓ **VLC** est un lecteur multimédia très léger qui permet de lire la plupart des fichiers audio et vidéo qui existent. L'avantage est qu'il contient lui-même les codecs nécessaires et donc vous n'avez rien à installer en plus. Il possède une interface personnalisable et permet également de lire le streaming et les DVD avec les sous-titres. De plus, il est entièrement libre, gratuit et multiplateforme. Il est donc disponible sur Windows et Linux. Ce lecteur possède en plus des fonctions intéressantes comme l'affichage images par images et le décalage des sous-titres. Vous pouvez le configurer de manière très complète [w5].
- ✓ **Spotify** est une plate-forme de distribution musicale à vocation communautaire dans la lignée de Last.fm, Deezer ou iTunes. Via une interface sobre et claire, elle propose une base de données musicale particulièrement bien étoffée ainsi qu'une fonction radio qui vous permettra de découvrir des titres de façon aléatoire. L'application offre un accès direct aux fiches des artistes et albums du site allmusic.com et s'intègre parfaitement au logiciel Last.fm pour le scrobbling (transmission des titres écoutés vers le site). De plus, elle conserve un historique des morceaux écoutés et permet de les partager grâce à l'envoi de liens par e-mail, messagerie instantanée ou directement sur Facebook ou Delicious [w6]. Pour l'installer sous ubuntu, il faut veiller à ajouter les dépôts [w7].
- ✓ **Audacity** est un outil pour la manipulation de données audio numériques. Audacity permet d'enregistrer du son numérique par le biais des entrées ligne/micro/cd des cartes sons. Il permet d'éditer (copier, coller, sectionner...) les sons sur plusieurs pistes, et il est accompagné de divers filtres et effets : pitch, tempo, réduction de bruit, égaliseur, filtres de Fourier, augmentation de fréquences précises, compression, amplification, normalisation, écho, phaser, wahwah, inversion... [w8].
- ✓ **MOM** est une bibliothèque musicale open source libre avec un player pour diffuser et organiser votre musique. Avec MOM, vous pourrez créer votre propre site musical avec espace utilisateurs, ceux ci

pourront créer des playlists [w9]. MOM utilise les Api de Last.fm, Youtube, Lyrics Wiki, LyricsFly. Ces caractéristiques essentielles sont :

- Bibliothèque musicale
  - Gestion des pochettes d'albums
  - Paroles des chansons
  - Visionneuse de videos Youtube
  - Identification utilisateur
  - Intégration Last.fm
  - Playlists
- ✓ **Audacious** est un logiciel pour gérer une bibliothèque audio, on pourra créer des listes de lecture, faire des tris, juste ce qu'il faut pour animer une soirée [w10].

### 2.6.3 Panorama des distributions Linux dédiés au multimédia

#### UBUNTU STUDIO

- ✓ -Distribution assez complète pour débiter et que je conseille, mais encore un petit peu lourde pour faire de la production musicale exigeante en temps réel (enregistrement et live), ce qui nécessite très peu de latence et de « XRuns » (microcoupures). Vous pouvez toutefois configurer tout ça par la suite. Pour le montage vidéo plus complet, KdenLive n'est pas installé par défaut, et son dépôt PPA pour avoir les toutes dernières versions non plus. En revanche il y a openshot pour du montage vidéo non-linéaire assez basique mais pas inutile. Installer et faire fonctionner correctement Cinelerra (montage video pro) vous donnera probablement pas mal de soucis...
- Les réglages du moteur audio Jackd sont déjà fait pour ne pas avoir a passer une heure avant de faire du son, c'est le but de la distribution. Un noyau Temps réel est aussi installé.
  - Les versions de logiciels sont celles d'ubuntu, donc pour avoir les versions les plus à jour il vous faudra rajouter à la main tous les dépôts PPA qui vous intéressent (alors que DreamStudio plus bas les a tous déjà prêts)
  - Pas mal de plugins audio LV2 et LADSPA déjà installés (ce sont des effets audio et midi).

- Pour encore plus de réglages spécifiques à vous d'installer les dépôt et paquets KXStudio qui amélioreront encore votre système pour la production musicale (Dépôts et logiciels installés par défaut avec DreamStudio plus bas)
- Le bureau par défaut est XFCE, plus léger que les autres bureaux Ubuntu comme Unity qui consomme trop de ressource [w11].

### DREAMSTUDIO

- ✓ DreamStudio n'est plus une distribution complète que l'on installe comme système d'exploitation Linux à part entière. Il s'agit désormais d'un ensemble de logiciel sous forme de « pack ».(test obsolète):
- ✓ -Plus complet qu'Ubuntu Studio, tous les dépôts spécifiques à la production audio-vidéo sont déjà là (KSTudios, les PPA des toutes dernières versions stables pour Gimp, KdenLive, cinelerra j'en passe...) Pour le son le moteur audio Jackd avec son interface graphique est déjà bien configuré pour le temps réel, et il y a un noyau Low Latency adéquat (faible latence, très très proche du temps réel). Pulse audio et jack sont configurés pour se passer la main l'un l'autre et vous pouvez donc écouter le son d'une application audio normale qui passe par Pulse audio quand jack est lancé, sans changer les réglages de sortie de celle-ci. Il y a vraiment beaucoup de logiciels installés et énormément de plugins LV2, LADSPA, DDSI pour faire de la musique avec l'embarras du choix.
- Par contre la version standard installe le bureau Unity, une mauvaise idée. On s'y habitue, mais c'est quand même pénible à utiliser quand on a pas envie d'avoir l'impression d'être sur une tablette plutôt qu'un ordinateur complet, et en plus il y a un gestionnaire graphique (compiz) qui tourne en permanence en tâche de fond, donc si votre souci est de réduire à fond les tâches pour faire du temps réel vraiment propre, vous aurez ça en plus à supporter. La solution c'est de prendre la version d'installation avec le bureau XFCE ou LXDE, ou d'installer un de ces bureaux après-coup et de ne plus démarrer sur le bureau de départ, Unity. Mais pour ces bureaux là, le développeur de DREAMSTUDIO ne garantie rien, c'est quand même un peu gênant. Pour ma part j'ai installé XFCE à la place d'Unity et après de multiples configurations j'en suis content, même s'il m'a fait des choses bizarres parfois avec pulse audio et Jackd [17].

### AVLINUX

- Très bonne distribution audio-vidéo, avec des performances excellentes pour la production musicale grâce au système Debian sur lequel elle repose. C'est une des seules à ne pas être basée sur Ubuntu (avec aussi Musix



qui n'est plus mise à jour?). Cette distribution peut vous permettre d'avoir une très bonne station de travail audio et production musicale, et de faire du montage vidéo HD et Full HD. Si vous supportez cinelerra ce logiciel est installé aussi.

-Un des objectifs affichés de la distribution est de pouvoir être installée sur des ordinateurs plus tous jeunes et d'en tirer le meilleur, car même des ordinateurs qui ont 10 ans sont capables de servir de station de travail audio-vidéo (si on n'exagère pas sur les effets), avec [les clips intermédiaires](#) notamment.

-Il ya beaucoup de préréglages pour faire fonctionner du matériel tel que des cartes sons USB dès le branchement des appareils (« Out Of The Box »), et c'est vraiment très agréable!

-Le DVD est rempli de la plupart des plugins imaginables, LV2, LADSPA, DSSI (mais pas de VST natif linux je crois), et il y aussi des plugin démo tout à fait utilisable en production de logiciels payants tels que piano pour avoir un beau son de piano acoustique.

-Le bureau est LXDE (changement: XFCE sur les dernières versions de 2014), le plus léger environnement graphique pour des performances accrues. [17].

- **Références bibliographique**

[1] [http://www.polytech-grenoble.fr/IMG/pdf/fiche\\_ricm.pdf](http://www.polytech-grenoble.fr/IMG/pdf/fiche_ricm.pdf)

[2] **Yvan KALIA**, Cours\_se1\_uvs, support de cours (2014)

[3] **Yvan KALIA**, Cours\_se2\_uvs, support de cours (2015)

[3] **Sylvain Baudry**, Introduction `a Unix et `a la programmation Shell, ESME-Sudriaversion 4.0, Jeffrey E. F. Friedl Mastering Regular Expressions, Powerful Techniques for Perl and Other Tools – 1st Edition January 1997 – O'Reilly & Associates Inc

[4] **Benmoussa Yahia**, support 35\_cours\_6\_getsion\_de\_la\_memoire\_v1.pdf, Université M'hamed Bougara de Boumerdès

[5] **GAUTHIER CATTEAU, ARMANDO MARTINS**, support, IntroductionLinux.pdf

- **Références bibliographiques**

[w1] <http://www.imedias.pro/cours-en-ligne/informatique/ordinateur/composants-ordinateur/>

[w2] <https://doc.ubuntu-fr.org/tar>

[w3] <http://www.xanetiz.com/vlc-lecteur-incontournable.html>