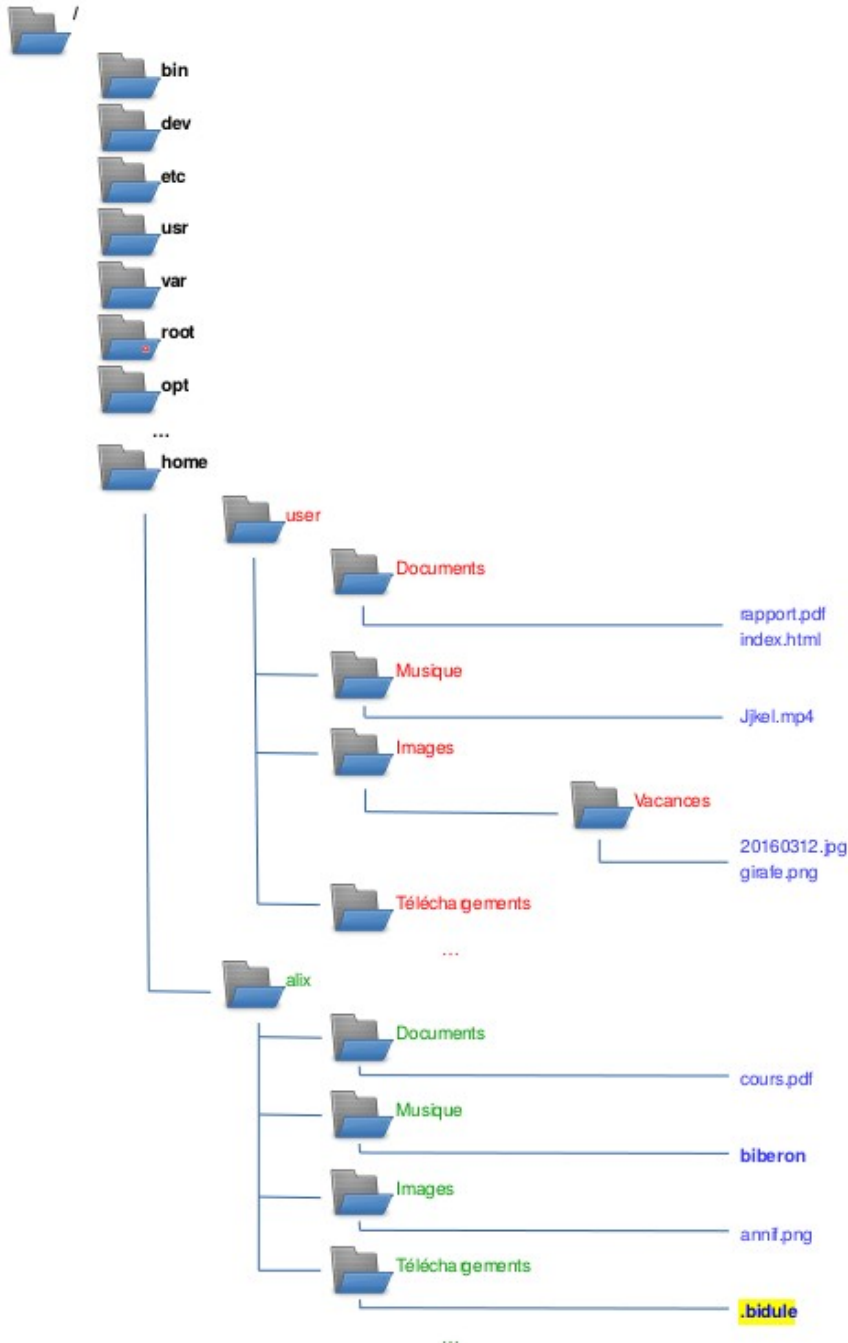


# CH0.1 PRISE EN MAIN D'UN SYSTÈME GNU/LINUX

## ACT 1 Donjon&Linux

## ACT 2 Gestion fichier Linux

### 1 Arborescence du système (exemple)



#### Remarques :

`/` désigne le répertoire racine.

`/home/user` est le répertoire personnel de l'utilisateur « user » :  
« user » n'a pas accès à « alix »

Les fichiers commençant par un « . » sont des fichiers cachés.

Le nom d'un chemin (*path*) d'un fichier est la concaténation depuis la racine de tous les répertoires qu'il est nécessaire de traverser pour y accéder, chacun étant séparé par le caractère « / ». C'est un **chemin absolu**.

Un chemin absolu :

- démarre à la racine : commence par « / »
- ne contient pas de « . » ni de « .. »

Le nom de chemin peut aussi être **relatif** à sa position courante dans un répertoire. Vous pouvez accéder à un autre répertoire en spécifiant le chemin le plus court relatif à la situation actuelle de l'arborescence. On utilise souvent 2 entrées particulières de répertoire :

- « . » qui désigne le répertoire courant (actuel)
- « .. » qui désigne le répertoire parent au répertoire actuel

Un chemin absolu :

- démarre à la racine : commence par « / »
- ne contient pas de « . » ni de « .. »

Le nom de chemin peut aussi être **relatif** à sa position courante dans un répertoire. Vous pouvez accéder à un autre répertoire en spécifiant le chemin le plus court relatif à la situation actuelle de l'arborescence. On utilise souvent 2 entrées particulières de répertoire :

- « . » qui désigne le répertoire courant (actuel)
- « .. » qui désigne le répertoire parent au répertoire actuel

Un chemin absolu :

- démarre à la racine : commence par « / »
- ne contient pas de « . » ni de « .. »

Le nom de chemin peut aussi être **relatif** à sa position courante dans un répertoire. Vous pouvez accéder à un autre répertoire en spécifiant le chemin le plus court relatif à la situation actuelle de l'arborescence. On utilise souvent 2 entrées particulières de répertoire :

- « . » qui désigne le répertoire courant (actuel)
- « .. » qui désigne le répertoire parent au répertoire actuel

Un chemin absolu :

- démarre à la racine : commence par « / »
- ne contient pas de « . » ni de « .. »

Le nom de chemin peut aussi être **relatif** à sa position courante dans un répertoire. Vous pouvez accéder à un autre répertoire en spécifiant le chemin le plus court relatif à la situation actuelle de l'arborescence. On utilise souvent 2 entrées particulières de répertoire :

- « . » qui désigne le répertoire courant (actuel)
- « .. » qui désigne le répertoire parent au répertoire actuel

Le nom de chemin peut aussi être **relatif** à sa position courante dans un répertoire. Vous pouvez accéder à un autre répertoire en spécifiant le chemin le plus court relatif à la situation actuelle de l'arborescence. On utilise souvent 2 entrées particulières de répertoire :

« . »	qui désigne le répertoire courant (actuel)
« . . »	qui désigne le répertoire parent au répertoire actuel

Le nom de chemin peut aussi être **relatif** à sa position courante dans un répertoire. Vous pouvez accéder à un autre répertoire en spécifiant le chemin le plus court relatif à la situation actuelle de l'arborescence. On utilise souvent 2 entrées particulières de répertoire :

« . »	qui désigne le répertoire courant (actuel)
« . . »	qui désigne le répertoire parent au répertoire actuel

Le nom de chemin peut aussi être **relatif** à sa position courante dans un répertoire. Vous pouvez accéder à un autre répertoire en spécifiant le chemin le plus court relatif à la situation actuelle de l'arborescence. On utilise souvent 2 entrées particulières de répertoire :

« . »	qui désigne le répertoire courant (actuel)
« . . »	qui désigne le répertoire parent au répertoire actuel

**Exemples :**

Le chemin absolu au fichier "cours.pdf" : **/home/alix/Documents/cours.pdf**

Le chemin relatif au fichier "cours.pdf" :

- si le répertoire courant est **alix** : **./Documents/cours.pdf**

ou **Documents/cours.pdf**

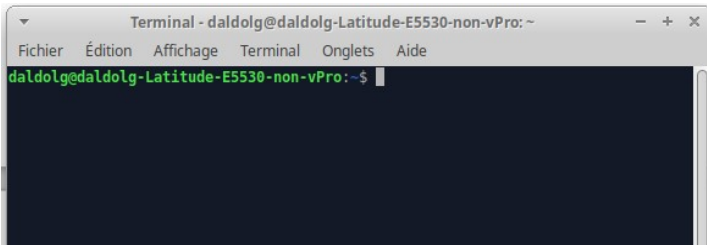
(Documents étant considéré existant dans le répertoire courant)

- si le répertoire courant est **Musique** : **../Documents/cours.pdf**

**Application :**

- Quel est le chemin absolu pour référencer le fichier « girafe.png » ?
- Si le répertoire courant est /home/user quel est le chemin relatif vers « index.html » ?
- Si le répertoire courant est /home/user/Images, quel est le chemin relatif vers « Jjkel.mp4 »?
- Si le répertoire courant est /home/user/Images/Vacances, quel est le chemin relatif vers «rapport.pdf » ?

## 2 Terminal et commandes de base



Console ou terminal

L'ouverture d'un terminal doit être conforme à l'image ci-contre, où l'on trouve :

- Le nom de l'utilisateur : **daldolg**
- Le nom de la machine : **daldolg-Latitude-E5530-non-vPro**
- Le nom du répertoire courant : ~ (tilde équivalent ici à **/home/user**)
- L'invite de commande : **\$**

Certaines commandes sont fastidieuses à taper, il faut bien le reconnaître. Pour vous aider :

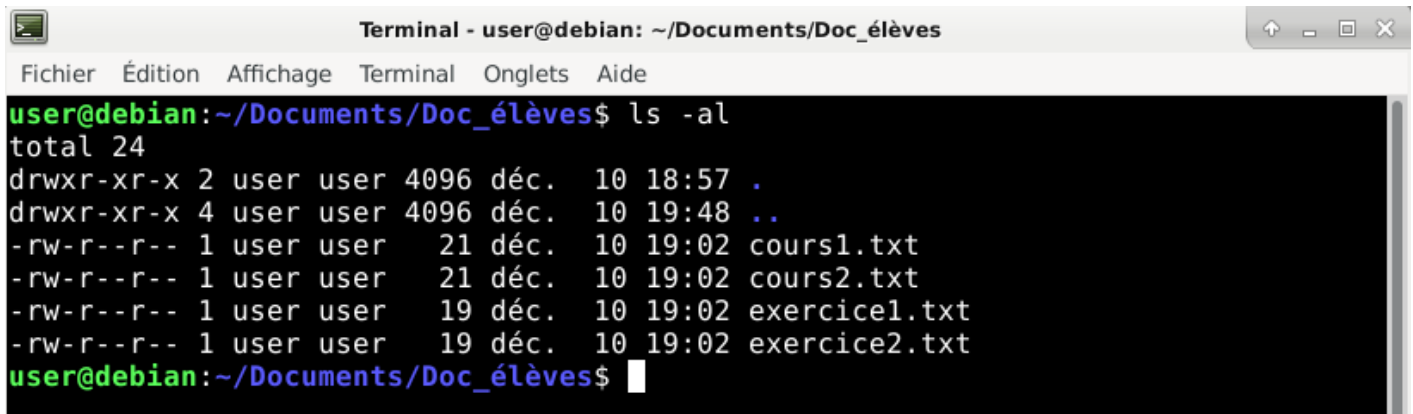
- La touche **Tab** active l'auto complétion
- Les flèches ↑ et ↓ rappellent les commandes précédentes qui sont mémorisées (même après arrêt de la machine)

**Applications :** Voir **TP Linux Commandes de base**

## 3 Gestion des droits des utilisateurs

Le rôle d'un système d'exploitation est aussi d'assurer la sécurité et l'accès aux données, ce qui est possible avec le mécanisme de droits.

Voilà ce que l'on peut apercevoir lorsque l'on tape la commande **ls -al**



Linux est un système d'exploitation multi-utilisateurs où le propriétaire d'un fichier peut partager l'accès à son fichier avec d'autre utilisateurs, membres d'un groupe, et en définir les types d'accès possibles. Lors de la création d'un compte utilisateur, un groupe du même nom est créé, et l'utilisateur créé fait partie de la liste des utilisateurs de ce groupe. Sur un système debian, chaque utilisateur appartient au moins à un groupe. Quand un utilisateur crée un fichier, il a tous les droits sur ce fichier.

Afin de permettre le partage de fichiers et faciliter le travail en équipe, Linux définit la notion de groupe d'utilisateurs. Le propriétaire d'un fichier pourra définir les différents types d'accès à son fichier pour lui même, et/ou les utilisateurs membres d'un groupe et/ou tous les autres utilisateurs (other).

<p>Les <b>types de fichiers</b>, définis en fonction des méthodes d'accès possibles pour les utiliser :</p> <ul style="list-style-type: none"> <li>- fichier ordinaire</li> <li><b>d</b> répertoire (directory)</li> </ul>	<p>Les <b>types d'utilisateurs</b> :</p> <ul style="list-style-type: none"> <li><b>u</b> ou <b>user</b> le propriétaire du fichier</li> <li><b>g</b> ou <b>group</b> les membres du groupe propriétaire du fichier</li> <li><b>o</b> ou <b>other</b> les autres utilisateurs du système</li> </ul>
<p>Les <b>autorisations d'accès</b> :</p> <ul style="list-style-type: none"> <li><b>r</b> Autorisation de lecture (<i>read</i>)</li> <li><b>w</b> Autorisation d'écriture (<i>write</i>)</li> <li><b>x</b> Autorisation d'exécution (<i>execute</i>)</li> </ul>	

**Exemple :** Analyse de la liste des fichiers de la figure ci-dessus

Pour les 2 premières lignes, il s'agit de répertoires, les autres lignes des fichiers.

En ce qui concerne le fichier « **exercice2.txt** » :

- Le propriétaire (user) a les droits de lecture et d'écriture (pas d'exécution, mais ce n'est pas un fichier exécutable)
- Les membres du groupe (user) ont le droit de lecture
- Les autres utilisateurs ont le droit de lecture

Les droits peuvent être modifiés grâce à la commande **chmod**. Les détails de cette commande sont visibles sur cette page [Openclassroom](#).

**Exemples utiles :**

commande	description
<b>chmod u+w fichier</b>	Ajouter les droits d'écriture au propriétaire (user, write)
<b>chmod g+r fichier</b>	Ajouter les droits de lecture au groupe du fichier (group, read)
<b>chmod o-x fichier</b>	Supprimer les droits d'exécution aux autres utilisateurs (other, execution)
<b>chmod a+rw dossier</b>	Ajouter les droits de lecture / écriture à tous (all)
<b>chmod -R a+rx dossier</b>	Ajouter les droits de lecture et d'exécution à tout ce que contient le

commande	description
	repertoire dossier (récursif).
<b>chmod -R 755 dossier</b>	Donner au propriétaire tous les droits (7xx), alors que seuls les droits de lecture et d'accès seront donnés aux autres (x55). Grace à l'option -R ces droits seront appliqués à tous les fichiers et dossiers contenus dans ce répertoire.

**Applications :**      Voir **Exercice Droits Utilisateurs (facultatif)**

## **4 Installer / désinstaller une application**

Debian GNU/Linux utilise le principe des dépôts en ligne pour fournir les applications. Ce principe permet de centraliser les logiciels et vous permet d'utiliser des interfaces simples pour administrer et/ou faire évoluer votre système : vous n'avez pas besoin de visiter les sites de logiciels.

### **4.1 Gestion des paquets APT**

Le logiciel de gestion de paquets **APT** (**A**dvanced **P**ackage **T**ool) est disponible directement depuis la ligne de commande, vous permettant l'administration fine de votre système.

Ces commandes nécessitent les droits de l'administrateur **root** (on commence par **sudo**) car elles affectent votre système.

commande	description
<b>apt update</b>	Mise à jour des métadonnées des dépôts.
<b>apt install mon_paquet</b>	Installation du paquet <i>mon_paquet</i> avec ses dépendances.
<b>apt upgrade</b>	Mise à jour sécurisée des paquets installés.
<b>apt dist-upgrade</b>	Mise à jour des paquets installés en installant ou supprimant d'autres paquets si nécessaire.
<b>apt remove mon_paquet</b>	Supprimer le paquet <i>mon_paquet</i> en laissant ses fichiers de configuration.
<b>apt autoremove</b>	Supprimer les paquets installés automatiquement lorsqu'ils ne sont plus nécessaires.
<b>apt purge mon_paquet</b>	Purger le paquet <i>mon_paquet</i> ainsi que ses fichiers de configuration.
<b>apt clean</b>	Nettoyer le cache local des paquets installés.
<b>apt autoclean</b>	Nettoyer le cache des fichiers des paquets périmés.

### Application :

Installez le paquet **gparted** qui permet de gérer les partitions du système.

Vous remarquerez que le gestionnaire propose d'installer d'autres paquets nécessaires à celui que vous souhaitez.

## **4.2 Les packages DEB**

La distribution Debian propose un outil de gestion de package permettant d'automatiser l'installation, la configuration et la mise à jour des logiciels installés par ce biais. Les packages Debian sont ainsi des fichiers portant l'extension **.deb** et pouvant être installés manuellement grâce à la commande suivante :

**dpkg -i nom\_du\_package.deb**

Pour désinstaller

**dpkg -r nom\_du\_package.deb**

Une commande pour supprimer aussi les fichiers de configuration du logiciel :

**dpkg -P nom\_du\_package.deb**

### Application :

Récupérez sur Internet , le package **filius\_1.7.4\_all.deb** et l'installer

## **5 Aller plus loin... Mais pas trop**

Toutes ces opérations étant fastidieuses, on peut créer un script qui automatise les processus.

Pour cela, il suffit :

- d'écrire dans un fichier texte commençant par **#!/bin/sh** (le [shebang](#)) avec toutes les opérations à effectuer.
- de le rendre exécutable avec la commande **chmod +x mon\_script**

### Exemple :

Script pour l'installation du logiciel arduino par **APT** et Filius par la méthode **wget** :

```
#!/bin/sh
sudo apt install arduino
wget https://www.lernsoftware-filius.de/downloads/Setup/filius_1.9.0_all.deb
filius_1.9.0_all.deb
```