

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

## 1 Historique GNU/LINUX

**1964** : MIT (Massachusetts Institute of Technology, Bell Labs de AT&T, General Electric → Développement du projet MULTICS (MULTiplexed Information and Computing Service) qui devra permettre de :

- Pouvoir être utilisé par plusieurs personnes à la fois ;
- Pouvoir lancer des traitements en tâche de fond ;
- Accroître la sécurité.

**1969** : Résultats du projet décevant → retrait de AT&T

- Poursuite du projet par Ken THOMPSON, Dennis RITCHIE, Rudd CANADAY du laboratoire Bell Labs sans soutien financier : renommage du projet UNICS (Uniplexed Information and Computing Service), puis UNIX.
- Retour de AT&T pour le financement

**1970** : Première utilisation officielle de la dénomination UNIX OS (Operating System).

**1973** : Réécriture d'UNIX en langage C pour améliorer sa portabilité.

**1974** : Diffusion sous licence peu restrictive. Les universités américaines contribuent fortement à la diffusion et son amélioration.

**1978** :

- UNIX V7
- AT&T rend la licence d'UNIX plus restrictive.
- Les universités dont celle de Californie BERKELEY poursuivent leurs développements avec des versions antérieures → (BSD : Berkeley Software Distribution) et publie sous licence Open Source.

**1980-1994** : C'est la guerre ... des versions d'UNIX (commerciale AT&T, Open Source BSD)

**1983** : Création du projet GNU (GNU is Not Unix) par Richard STALLMAN

**1985** : Création de la Free Software Foundation par Richard STALLMAN

**1988** :

- Standardisation de UNIX (tout ce que UNIX doit contenir : commandes de base, interpréteur de commandes, interfaces utilisateur, ...).
- Norme IEEE 1003 aussi appelée POSIX (Portable Operating System Interface et le X ... pour UNIX !).
- Tout éditeur de système UNIX (commercial ou gratuit) désirant assurer une compatibilité avec l'ensemble des UNIX doit implémenter ce standard (la programmation restant libre)

**1991** : Création du noyau LINUX par Linus TORVALDS. Les autres outils nécessaires ont déjà été développés par le projet GNU

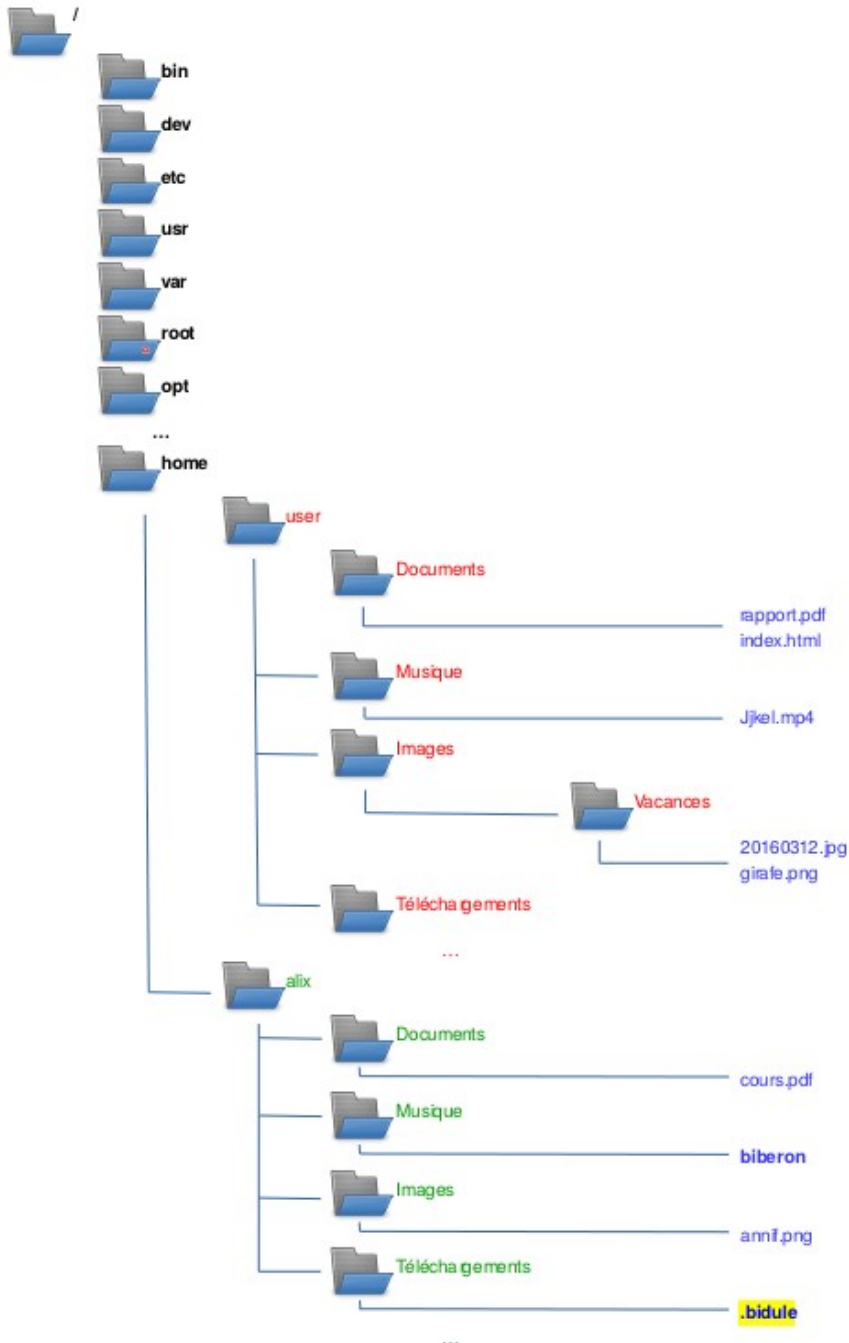
**1992** : Fonctionnement de l'interface graphique X11

**1994-1997** : Apparition des grandes distributions de LINUX (Red Hat, Debian, SuSE ...)

**1996** : Création de la mascotte Tux

**2011** : noyau 3.0. Linux à 20 ans

## 2 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 n'ont pas forcément une extension

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

### **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**

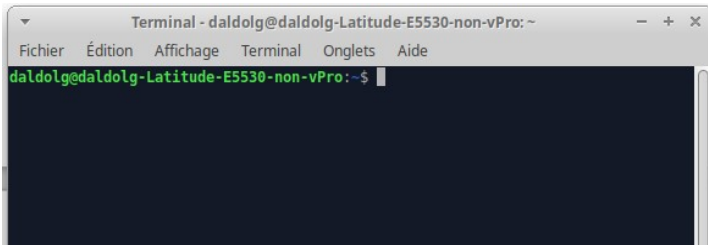
*(Documents étant considéré existant dans le répertoire courant)*

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

### **Exercice 1 :**

- 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 » ?

### 3 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)

#### **Exercice 2 :**

L'objectif est de créer l'arborescence décrite par l'image ci-dessous en ligne de commande. On va plus ou moins volontairement commettre quelques erreurs pour apprendre comment les rectifier ...

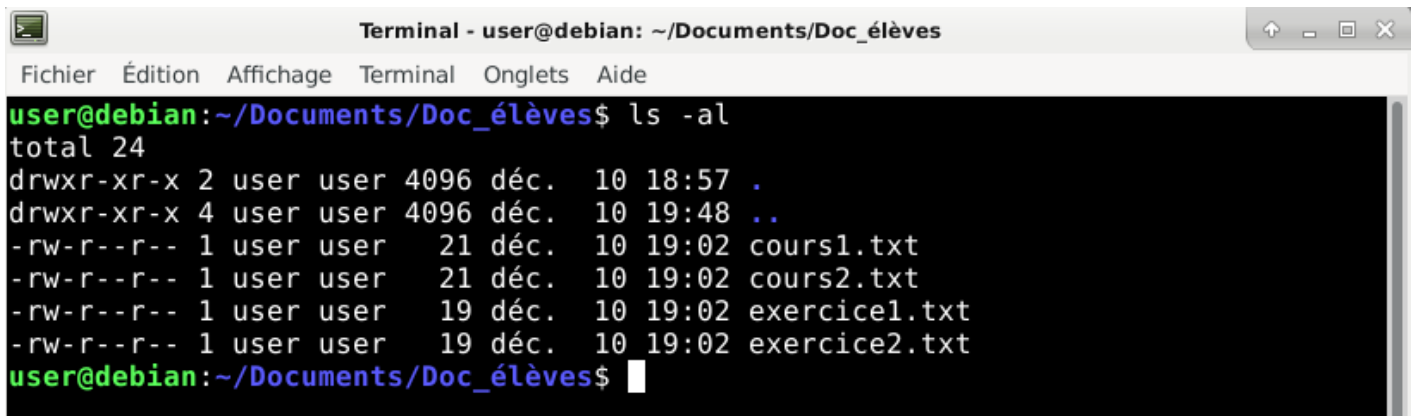
***(Voir document à part)***

#### **Bilan des commandes :**


## 4 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**



```
Terminal - user@debian: ~/Documents/Doc_élèves
Fichier  Édition  Affichage  Terminal  Onglets  Aide
user@debian:~/Documents/Doc_élèves$ ls -al
total 24
drwxr-xr-x 2 user user 4096 déc. 10 18:57 .
drwxr-xr-x 4 user user 4096 déc. 10 19:48 ..
-rw-r--r-- 1 user user  21 déc. 10 19:02 cours1.txt
-rw-r--r-- 1 user user  21 déc. 10 19:02 cours2.txt
-rw-r--r-- 1 user user  19 déc. 10 19:02 exercice1.txt
-rw-r--r-- 1 user user  19 déc. 10 19:02 exercice2.txt
user@debian:~/Documents/Doc_élèves$
```

Linux est un système d'exploitation multi-utilisateurs où le propriétaire d'un fichier peut partager l'accès à son fichier avec d'autres 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).

Les différents **types de fichiers** sont définis en fonction des méthodes d'accès possibles pour les utiliser :

- - fichier ordinaire
- d répertoire (directory)

Il existe **trois types d'utilisateurs** :

- u user Le propriétaire du fichier
- g group les membres du groupe propriétaire du fichier
- o other les autres utilisateurs du système

Pour chaque fichier et pour chaque type d'utilisateur **il y a trois autorisations d'accès** :

- r : Autorisation de lecture (read)
- w : Autorisation d'écriture (write)
- x : Autorisation d'exécution (execute)

**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

Correspondances de représentation des droits		
Droit	Valeur alphanumérique	Valeur octale
aucun droit	—	0
exécution seulement	-x	1
écriture seulement	-w-	2
écriture et exécution	-wx	3
lecture seulement	r—	4
lecture et exécution	r-x	5
lecture et écriture	rw-	6
tous les droits (lecture, écriture et exécution)	rwX	7

Exemples de commande

**chmod u+w fichier** Ajouter les droits d'écriture au propriétaire (user, write)

**chmod g+r fichier** Ajouter les droits de lecture au groupe du fichier (group, read)

**chmod o-x fichier** Supprimer les droits d'exécution aux autres utilisateurs (other, execution)

**chmod a+rw dossier** Ajouter les droits de lecture / écriture à tous (all)

**chmod -R a+rx dossier** Ajouter les droits de lecture et d'exécution à tout ce que contient le répertoire dossier (récursif).

**chmod 764 dossier** Tous les droits pour le propriétaire (7xx), lecture et écriture pour le groupe (x6x) et lecture uniquement pour les autres (xx4).

**chmod -R 755 dossier** Donner au propriétaire tous les droits (7xx), alors que seuls les droits de lecture et d'accès seront donnés aux autres (x55). Grâce à l'option -R ces droits seront appliqués à tous les fichiers et dossiers contenus dans ce répertoire.

**Exercice 3 :** (Voir document à part)

## 5 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.

Les adresses internet des dépôts Debian sont inscrites dans le fichier **/etc/apt/sources.list** et les fichiers du type **/etc/apt/sources.list.d/xxx.list**.

### 5.1 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.

#### → Les commandes « utilisateur » de recherche et d'information

Ces commandes peuvent être lancées en tant que simple utilisateur car elles n'affectent pas votre système.

commande	description
<b>apt show <i>toto</i></b>	Afficher les informations du paquet <i>toto</i>
<b>apt search <i>toto</i></b>	Rechercher les paquets qui correspondent à <i>toto</i>
<b>apt-cache policy <i>toto</i></b>	Affiche les versions disponibles de <i>toto</i>

#### → Les commandes 'administrateur' de maintenance

Ces commandes nécessitent les droits de l'administrateur **root** car elles affectent votre système.

commande	description
<b>apt update</b>	Mise à jour des métadonnées des dépôts.
<b>apt install <i>toto</i></b>	Installation du paquet <i>toto</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 <i>toto</i></b>	Supprimer le paquet <i>toto</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 <i>toto</i></b>	Purger le paquet <i>toto</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.

#### **Exercice 4 :**

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.

### **5.2 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**

#### **Exercice 5 :**

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

## **6 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
mdp = ## mon mot de passe
user = ## mon login
sudo apt install arduino <<< mdp

wget https://www.lernsoftware-filius.de/downloads/Setup/filius_1.9.0_all.deb
filius_1.9.0_all.deb
```

La commande **<<<** permet de « rentrer » une variable dans la commande précédente.