



# TUX101

Ou pourquoi `sudo rm -rf /` est une mauvaise idée

Thomas PUJOL

[thomas.pujol@telecom-paris.fr](mailto:thomas.pujol@telecom-paris.fr)

13 septembre 2023



# Plan

## Histoire de GNU/Linux

### Linux

- Le shell

- Généralités

- Système de fichiers

- Gestion des droits

- Manipulation des dossiers

- Manipulation des fichiers

- Administration système

### SSH

### Bonus



## Qu'est-ce que Linux ? Et GNU ?

- **Linux** est un noyau (kernel) open-source



Fun fact : Linus Torvalds a développé Git en une semaine.

Figure – Linus Torvalds



## Qu'est-ce que Linux ? Et GNU ?

- **Linux** est un noyau (kernel) open-source



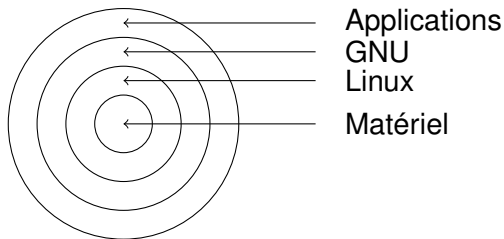
Figure – Linus Torvalds

# Qu'est-ce que Linux ?

## Et GNU ?

### ■ **Linux** est un noyau (kernel) open-source

- Un noyau est le cœur d'un système d'exploitation
- Gestion des ressources (mémoire, des processus, des périphériques, etc.)
- Assure la communication entre les parties matérielles et logicielles



**Figure** – Architecture d'un système d'exploitation GNU/Linux

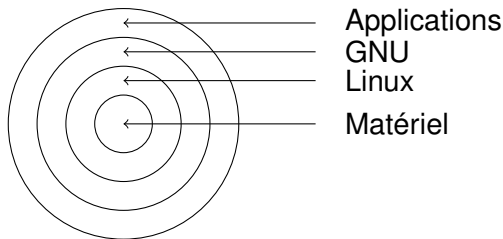


# Qu'est-ce que Linux ?

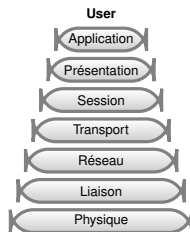
## Et GNU ?

### ■ **Linux** est un noyau (kernel) open-source

- Un noyau est le cœur d'un système d'exploitation
- Gestion des ressources (mémoire, des processus, des périphériques, etc.)
- Assure la communication entre les parties matérielles et logicielles



**Figure – Architecture d'un système d'exploitation GNU/Linux**



**Figure – Modèle OSI**



# Qu'est-ce que Linux ?

## Et GNU ?

### ■ **Linux** est un noyau (kernel) open-source

- Un noyau est le cœur d'un système d'exploitation
- Gestion des ressources (mémoire, des processus, des périphériques, etc.)
- Assure la communication entre les parties matérielles et logicielles

### ■ **GNU** : GNU's Not Unix

- Un système d'exploitation
- Un ensemble de logiciels
- Un ensemble de bibliothèques



# L'histoire de GNU

## GNU : GNU's Not Unix



Richard Stallman

- Créé par Richard Stallman en 1983
- Système reprenant les concepts d'Unix
  - Unix : OS créé par AT&T en 1969
  - Développé par Bell Labs (AT&T), développement dirigé par Kenneth Thompson
  - Repose sur le `shell` (ligne de commande ou interpréteur)
- Exemples d'outils contenus dans GNU :
  - `gcc` (GNU Compiler Collection) : compilateur C
  - `make` : outil de construction
  - `macchanger` (GNU MAC Changer) : utilitaire qui facilite la manipulation des adresses MAC des interfaces réseau
  - Pour plus d'outils, voir : <https://www.gnu.org/software/>





## Attaque la suite, on s'ennuie

- GNU/Linux est considéré comme un système d'exploitation
- Par abus de langage, GNU/Linux est rangé dans la famille Unix (Unix-like)
- On abrège usuellement GNU/Linux par Linux
- Pour participer au développement de Linux : <https://www.kernel.org/>
- Pour participer au développement de GNU : <https://www.gnu.org/help/help.html>



# Plan

Histoire de GNU/Linux

Linux

- Le shell

- Généralités

- Système de fichiers

- Gestion des droits

- Manipulation des dossiers

- Manipulation des fichiers

- Administration système

SSH

Bonus



# Plan

Histoire de GNU/Linux

Linux

- Le shell

- Généralités

- Système de fichiers

- Gestion des droits

- Manipulation des dossiers

- Manipulation des fichiers

- Administration système

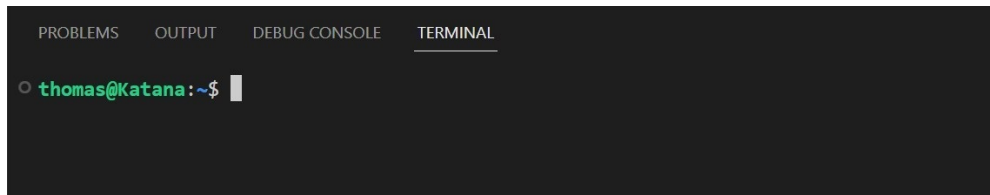
SSH

Bonus



# Le shell

## Ligne de commande



- Le shell est un programme, ici `bash` (GNU) qui permet d'interagir avec le système d'exploitation
- Permet de lancer des programmes, de créer des fichiers, de les déplacer, de les supprimer, etc.
- Permet de lancer des programmes en mode interactif, non-interactif
- Permet de lancer des programmes en mode graphique, ou non



# Plan

Histoire de GNU/Linux

Linux

Le shell

Généralités

Système de fichiers

Gestion des droits

Manipulation des dossiers

Manipulation des fichiers

Administration système

SSH

Bonus



# Un programme

Très utile

## `man` : **Manual**

- `man <program_name>` : affiche le manuel du programme  
`<program_name>`
- `man -k <keyword>` : affiche les programmes dont le manuel contient le mot-clé `<keyword>`

Note : Vous pouvez obtenir la documentation des appels systèmes.

Exemple : `man fork`



# À l'aide !

SOS d'un Télécomien en détresse

## Pour obtenir de l'aide pour utiliser un programme

- Un helper est disponible
- On l'appelle avec `<program_name> --help`



# Les variables d'environnement

## Et le PATH

- Les variables d'environnement sont des variables qui sont accessibles par tous les programmes
- Elles sont modifiables avec la commande `set`
- Elles sont accessibles avec la commande `env`

### PATH

- Variable d'environnement
- Liste des répertoires où se trouvent les programmes
- Exemple : `PATH=/usr/local/bin:/usr/bin:/bin`





# Plan

Histoire de GNU/Linux

Linux

Le shell

Généralités

**Système de fichiers**

Gestion des droits

Manipulation des dossiers

Manipulation des fichiers

Administration système

SSH

Bonus



# Un système de fichiers

## Différent de Windows

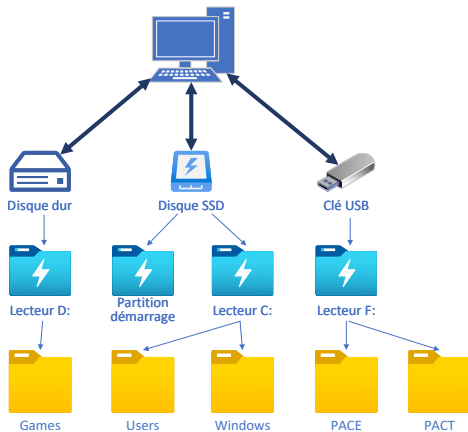


Figure – Arborescence Windows

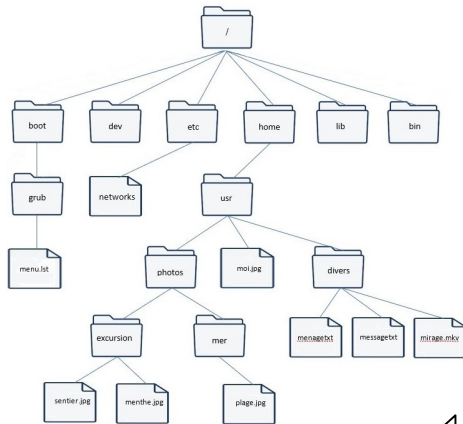


Figure – Arborescence Linux



# Un système de fichiers

## Différent de Windows

- Dans Linux, **TOUT EST FICHIER** (ou dossier)
- Les périphériques, les ports, les connexions, etc. sont des **fichiers**
- Les fichiers contenus dans `/dev` sont des périphériques
- Exemples :
  - `/dev/sda` : disque A
  - `/dev/sda1` : partition 1 du disque A
  - `/dev/input/*` : périphériques d'entrée
  - `/dev/tty` : ports série
  - `/dev/input/event1` : souris



# Un système de fichiers

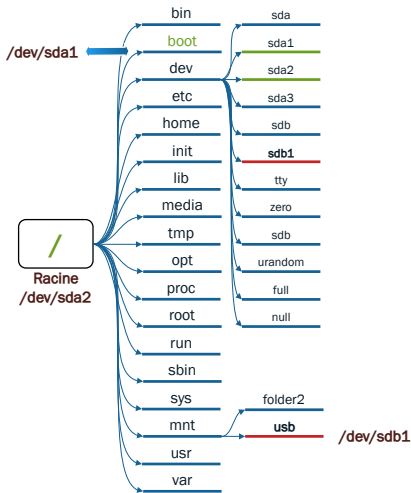
## Différent de Windows

- Dans Linux, **TOUT EST FICHIER** (ou dossier)
- Les périphériques, les ports, les connexions, etc. sont des **fichiers**
- Les fichiers contenus dans `/dev` sont des périphériques
- Exemples :
  - Je branche une clé USB formatée en FAT32 à mon PC
  - Mon PC uniquement un disque SSD branché : `/dev/sda`
  - Ma clé USB est détectée et apparaît : `/dev/sdb`
  - Nous avons sa partition 1 (en FAT32) : `/dev/sdb1`
  - Nous pouvons la monter : `mount /dev/sdb1 /mnt/usb`



# Montage d'une partition sur un système Linux

```
mount /dev/sdb1 /mnt/usb
```



- `/dev/sdb1` : partition 1 du disque B
- `/mnt/usb` : dossier de montage
- `mount` : commande de montage
- `umount` : commande de démontage
- Systèmes de fichiers propres à Linux : `ext2` , `ext3` , `ext4`



# Plan

## Histoire de GNU/Linux

## Linux

Le shell

Généralités

Système de fichiers

**Gestion des droits**

Manipulation des dossiers

Manipulation des fichiers

Administration système

## SSH

## Bonus



# Gestion des droits

Une vraie !

```
thomas@Katana:~$ ls -la
total 128
drwxr-xr-x 13 thomas thomas 4096 Feb 27 15:49 .
drwxr-xr-x  3 root   root   4096 Nov  1 22:42 ..
drwxr-xr-x  7 thomas thomas 4096 Dec  6 23:41 INF104
```

■ **Propriétaire**

■ **Groupe**

■ **Autres**

■ **Lecture** : `r`

■ **Écriture** : `w`

■ **Exécution** : `x`

■ **Changement de droits** : `chmod`

■ **Changement de propriétaire** : `chown`

■ **Changement de groupe** : `chgrp`



# Gestion des droits

## Groupes et notation octale

```
thomas@Katana:~$ id
uid=1000(thomas) gid=1000(thomas) groups=1000(thomas),4(adm),20(dialout),24(cdrom),25(floppy),27(sudo),29(audio),30(dip),44(video),46(plugdev),117(netdev),999(docker)
```

- Chaque utilisateur peut appartenir à plusieurs groupes
- Groupes : gestion des droits simplifiée
- Notation des droits
  - Symbolique : `r` , `w` , `x` : droits de lecture, écriture et exécution
  - Octale : `4` , `2` , `1` : droits de lecture, écriture et exécution
    - Pour déterminer les droits octaux, on additionne les droits correspondants
    - `rw``x` = `4+2+1` = `7`
    - `rx` = `4+1` = `5`





# Plan

Histoire de GNU/Linux

Linux

Le shell

Généralités

Système de fichiers

Gestion des droits

**Manipulation des dossiers**

Manipulation des fichiers

Administration système

SSH

Bonus



# Navigation dans l'arborescence

Il ne suffit pas de fuir, il faut fuir dans le bon sens.

## cd : Change Directory

- `cd` ou `cd ~` : se rend au dossier personnel
- `cd ..` : se rend au dossier parent
- `cd ../coucou` : se rend au dossier `coucou/` contenu dans le parent
- `cd /` : se rend à la racine
- `cd /home/thomas` : se rend au dossier `/home/thomas`
- `cd /var/www/html` : se rend au dossier `/var/www/html`



# Navigation dans l'arborescence

Naviguer à vue, c'est retrouver le sel et le sens de la vie.

## `pwd` : Print Working Directory

- `pwd` : affiche le chemin absolu du dossier courant



# Navigation dans l'arborescence

Je ne crois pas qu'il y ait de bonne ou de mauvaise situation.

## `ls` : List

- `ls` : affiche le contenu du dossier courant
- `ls -l` : affiche le contenu du dossier courant avec les droits
- `ls -a` : affiche le contenu du dossier courant avec les fichiers cachés
- `ls -la` : affiche le contenu du dossier courant avec les droits et les fichiers cachés

En spécifiant un dossier en paramètre, `ls` affiche le contenu de ce dossier.



# Opérations dans l'arborescence

Dépêchons-nous je n'ai que quelques heures.

## `mkdir` : Make Directory

- `mkdir <dir_name>` : crée un dossier
- `mkdir -p <dir_name>` : crée un dossier et tous les dossiers parents

Exemple : En admettant que le dossier `/var/www/html` existe et soit vide, `mkdir -p /var/www/html/demo/test` crée deux dossiers : `demo/` dans `/var/www/html/` et `test/` dans le dossier `/var/www/html/demo/`.



# Plan

Histoire de GNU/Linux

Linux

Le shell

Généralités

Système de fichiers

Gestion des droits

Manipulation des dossiers

**Manipulation des fichiers**

Administration système

SSH

Bonus



# Opérations sur les fichiers

Si tu ne décides pas de ton destin c'est ton destin qui décide pour toi.

## `touch` : Touch

- `touch <filename>` : crée un fichier
- `touch -a <date> <filename>` : modifie la date d'accès du fichier
- `touch -m <date> <filename>` : modifie la date de modification du fichier



## Opérations dans l'arborescence

**cat** : **Concatenate**

- `cat <filename>` : affiche le contenu du fichier





## Opérations dans l'arborescence

### `find` : Find

- `find <dir_name>` : affiche le contenu du dossier et de ses sous-dossiers
- `find <dir_name> -name <filename>` : affiche le contenu du dossier et de ses sous-dossiers dont le nom correspond à `<filename>`
- `find <dir_name> -type <type>` : affiche le contenu du dossier et de ses sous-dossiers dont le type correspond à `<type>`



# Opérations sur l'arborescence

D'une pensée géniale on peut supprimer tous les mots.

## `rm` : Remove

- `rm <filename>` : supprime un fichier
- `rm -r <dir_name>` : supprime un dossier et son contenu (récursivement)
- `rm -f <filename>` : supprime un fichier sans confirmation
- `rm -rf <dir_name>` : supprime un dossier (récursivement) sans confirmation



# Opérations sur l'arborescence

Qui n'imité point n'invente point.

## `cp` : Copy

- `cp <filename> <new_filename>` : copie un fichier
- `cp -r <dir_name> <new_dir_name>` : copie un dossier et son contenu (récursivement)



# Opérations sur l'arborescence

La vie est un mystère qu'il faut vivre, et non un problème à résoudre.

## mv : Move

- `mv <filename> <new_filename>` : renomme un fichier
- `mv <filename> <dir_name>` : déplace un fichier dans un dossier
- `mv <dir_name> <new_dir_name>` : renomme un dossier
- `mv <dir_name> <dir_name2>` : déplace un dossier dans un autre dossier



# Opérations sur l'arborescence

## Un fil rouge invisible

### ln : Link

- `ln -s <filename> <link_name>` : crée un lien symbolique



# Un autre programme bien sympathique

La vie est un mystère qu'il faut vivre, et non un problème à résoudre.

## `grep` : Global Regular Expression Print

- `grep <pattern> <filename>` : affiche les lignes du fichier qui contiennent le motif `<pattern>`
- `grep -r <pattern> <dir_name>` : affiche les lignes des fichiers du dossier et de ses sous-dossiers qui contiennent le motif `<pattern>`



# Plan

Histoire de GNU/Linux

Linux

- Le shell

- Généralités

- Système de fichiers

- Gestion des droits

- Manipulation des dossiers

- Manipulation des fichiers

- Administration système**

SSH

Bonus



# Exécuter en tant qu'administrateur

With great power comes great responsibility

## `sudo` : Super User Do

- `sudo <command>` : exécute la commande en tant qu'administrateur
- `sudo -i` : ouvre un shell en tant qu'administrateur





# Créer un utilisateur ou le supprimer

## `useradd` : User Add

- `useradd <username>` : crée un utilisateur
- `useradd -m <username>` : crée un utilisateur et son dossier personnel
- `useradd -m -s <shell> <username>` : crée un utilisateur et son dossier personnel avec un shell spécifique
- `userdel <username>` : supprime un utilisateur



## Modifier son mot de passe

`passwd` : **Password**

- `passwd <username>` : modifie le mot de passe d'un utilisateur
- `passwd` : modifie le mot de passe de l'utilisateur courant



# Créer, modifier un groupe ou le supprimer

## `groupadd` : Group Add

- `groupadd <groupname>` : crée un groupe
- `usermod -a -G <groupname> <username>` : ajoute un utilisateur à un groupe
- `deluser <username> <groupname>` : supprime un utilisateur d'un groupe
- `groupdel <groupname>` : supprime un groupe



# Gérer ses logiciels

Mais avant, parlons des distributions

## Distribution

- Ubuntu, Debian, Fedora, Arch Linux sont des distributions
- Distribution : ensemble de logiciels pré-installés
- **Ubuntu** : distribution basée sur Debian
- **Debian** : distribution basée sur **GNU/Linux**
- Ici nous ne parlerons que de Debian.



# Gérer ses logiciels

Ici on appelle ça les paquets

`apt-get` ou `apt` : **Advanced Packaging Tool**

- `apt-get update` : met à jour la liste des paquets disponibles

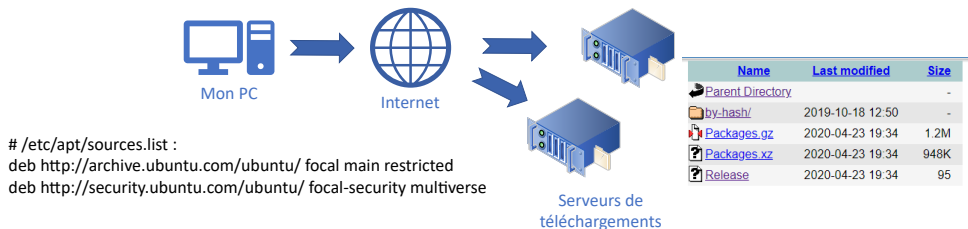


Figure – Fonctionnement d'APT



# Gérer ses logiciels

Ici on appelle ça les paquets

## `apt-get` ou `apt` : **Advanced Packaging Tool**

- `apt-get update` : met à jour la liste des paquets disponibles
- `apt-get upgrade` : met à jour les paquets installés
- `apt-get install <package_name>` : installe un paquet
- `apt-get remove <package_name>` : supprime un paquet
- `apt-get purge <package_name>` : supprime un paquet et ses fichiers de configuration



# Gérer ses logiciels

## Quand on est un peu plus vieux

### `dpkg` : Debian Package

- `dpkg -i <package_name>` : installe un paquet
- `dpkg -r <package_name>` : supprime un paquet
- `dpkg -P <package_name>` : supprime un paquet et ses fichiers de configuration



# Plan

## Histoire de GNU/Linux

## Linux

Le shell

Généralités

Système de fichiers

Gestion des droits

Manipulation des dossiers

Manipulation des fichiers

Administration système

## SSH

## Bonus





# Chiffrement asymétrique

Clé publique / privée



Figure – Chiffrement asymétrique



# Chiffrement asymétrique

## Générer ses clés

### `ssh-keygen` : Secure Shell Key Generator

- `ssh-keygen -t rsa` : génère une clé RSA
- `ssh-keygen -t ed25519` : génère une clé Ed25519 (plus rapide)
- `ssh-keygen -t rsa -b 4096` : génère une clé RSA de 4096 bits
- Puis suivre les instructions



# Une invite de commande à distance

## Mais pas que !

### ssh : Secure Shell

- `ssh <user>@<host>` : se connecte à un serveur distant
- `ssh -p <port> <host>` : spécification d'un port
- `ssh <host> -i <key>` : connexion avec une clé spécifique
- `ssh -J host1,host2 host` : connexion via d'autres serveurs
- `ssh <host> -X <command>` : untrusted X11 forwarding
- `ssh <host> -Y <command>` : trusted X11 forwarding
- `ssh <host> -R <port>:<host>:<port>` : crée un tunnel SSH inversé
- `ssh <host> -L <port>:<host>:<port>` : crée un tunnel SSH



## Une invite de commande à distance

~/.ssh/config

### Fichier de configuration ~/.ssh/config

- Pour mémoriser les paramètres de connexion en fonction de l'hôte
- Host <host> : définit un hôte – NB : les tabulations sont importantes
- HostName <host> : définit l'adresse de l'hôte (obligatoire)
- Port <port> : définit le port de connexion
- User <user> : définit l'utilisateur à utiliser pour se connecter
- IdentityFile <key> : définit la clé à utiliser pour se connecter
- ProxyJump <host> : définit un serveur intermédiaire (-J)

Ce fichier est également fonctionnel sur Windows à l'emplacement :  
%USERPROFILE%/.ssh/config



# Une invite de commande à distance

## SSH Agent

### `ssh-agent` : **Secure Shell Agent**

- `ssh-agent` : démarre un agent SSH
- `ssh-add <key>` : ajoute une clé à l'agent SSH
- `ssh-add -l` : liste les clés ajoutées à l'agent SSH
- `ssh-add -D` : supprime toutes les clés de l'agent SSH



# Une invite de commande à distance

## SCP

### scp : Secure Copy

- `scp <file> <user>@<host>:<path>` : copie un fichier vers un serveur distant
- `scp <user>@<host>:<file> <path>` : copie un fichier depuis un serveur distant
- `scp -r <dir> <user>@<host>:<path>` : copie un dossier vers un serveur distant
- `scp -P <port> <file> <host>:<path>` : spécification d'un port
- `scp -i <key> <file> <host>:<path>` : connexion avec une clé spécifique



# Plan

## Histoire de GNU/Linux

## Linux

- Le shell

- Généralités

- Système de fichiers

- Gestion des droits

- Manipulation des dossiers

- Manipulation des fichiers

- Administration système

## SSH

## Bonus



# Édition de texte

## avec nano

**nano** : **Nano is ANOther editor** ou  $10^3$  **Pico**

- `nano <file>` : édite un fichier
- `nano -h` : affiche l'aide de nano

### Commandes lors de l'utilisation de nano

- `Ctrl + G` : affiche l'aide
- `Ctrl + O` : écrit le fichier avec un nom donné
- `Ctrl + S` : écrit le fichier sans changer le nom
- `Ctrl + C` : obtient la position du curseur
- `Ctrl + W` : recherche un mot
- `Ctrl + X` : quitte nano





# Édition de texte

## avec vim

### vim : Vi IMproved

- `vim <file>` : édite un fichier
- `vim -h` : affiche l'aide de vim

### Commandes lors de l'utilisation de vim

- `i` : insère du texte
- `a` : ajoute du texte
- `o` : ajoute une ligne
- `dd` : supprime une ligne
- `yy` : copie une ligne
- `p` : colle une ligne
- `u` : annule la dernière action
- `Ctrl + R` : rétablit la dernière action annulée
- `:w` : écrit le fichier
- `:q` : quitte vim
- `:wq` : écrit et quitte vim

