

INFORMATIQUE POUR LA ROBOTIQUE 2

Cours n °1 : Introduction à Linux

Enseignante : Maëva LECAVELIER - maeva.lecavelier@gmail.com

Sommaire

01

Linux

Qu'est-ce que c'est

02

Intérêt de Linux

Pourquoi Linux ?

03

Objectifs

Objectifs de ce cours

04

Utiliser Linux

Comment faire ?

01

Linux

Qu'est-ce que c'est ?

Qu'est-ce que Linux ?



Créé en 1991 par Linus Torvald

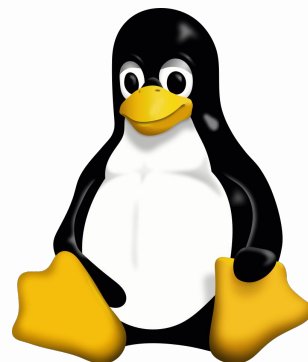


**C'est un système d'exploitation
(comme le sont Windows, MacOS,
Android...)**



C'est gratuit et open-source

Ça veut dire que tout le monde peut participer, corriger et améliorer ce qui est développé. Ce qui permet aussi de l'utiliser librement, et c'est pourquoi il existe plus de 500 distributions différentes (Ubuntu, Debian, Fedora, RedHat, ...)



02

Intérêt de Linux

Pourquoi utiliser Linux ?

Qui utilise Linux ?

- Android
- WSL (Windows Subsystem for Linux)
- Les infrastructures pour Internet (serveur, data center...)
- Les systèmes embarqués (Raspberry, Jetson Nano, Android Automotive...)
- Des grandes bourses (Londres, New-York...)
- Certains systèmes informatiques de villes
- Etc....

Quel intérêt d'utiliser Linux

- **Performance et sécurité** : grâce à l'héritage d'UNIX
- **Open source** : liberté d'utiliser, de copier, d'étudier et de modifier les logiciels ainsi que de redistribuer les versions modifiées. Assure une robustesse hors norme
- **Léger pour l'embarqué** : certaines distributions ne font que quelques MB
- Très bien **documenté** et **système ouvert** qui permet de manipuler le système au plus bas niveau

Objectifs du cours

Pourquoi apprendre Linux

Contexte

La robotique est un subtil mélange d'informatique, d'électronique et de mécanique

Il nous faut donc un "cerveau" capable de gérer tout ça

C'est là qu'intervient les plateformes embarquées, et dans ce cours nous utiliserons la **Jetson Nano reComputer j1010**, qui utilise...



Linux !

D'où l'importance de s'y
intéresser !

Objectif : ajouter de nouvelles fonctionnalités à votre robot



Automatisation



Scheduler



Puissance



Mémoire

04

Utiliser Linux

Comment faire

L'arborescence UNIX et donc Linux

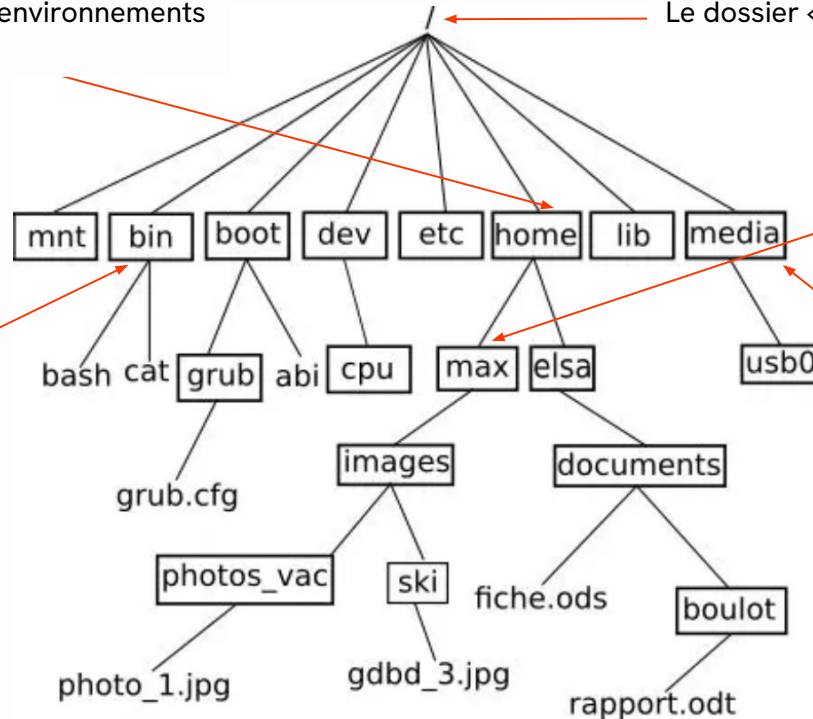
Le dossier qui contient tous les environnements des utilisateurs

Le dossier « racine » ou / : à l'origine de tout

Contient tous les binaires des commandes basiques utilisées par tous les utilisateurs

Pour Max, c'est l'équivalent du « Mes documents » de Windows

Contient le contenu de tous les médias USB, CD... branchés au système



Quelques commandes



Manipuler les fichiers et dossiers : commandes essentielles

ls, mkdir, cd, touch, vi, cp, mv, cat, echo



Gestion des droits et utilisateurs : commandes utiles

chmod, adduser, addgroup



Gestion de l'espace et des médias : commandes

avancées

du, df, lsusb, fdisk...

Automatisation : les scripts bash

Fichiers exécutables en .sh

Un script bash est un fichier contenant des lignes de commandes, avec éventuellement des boucles, des éléments logiques (and, or, not...). Ils permettent d'automatiser l'exécution de commandes

Commencent toujours par **#!/bin/bash**

Vous vous souvenez du dossier « bin » ? Ici, cette ligne permet de déclarer que le script va s'exécuter en utilisant l'environnement d'exécution « bash ». Le script ne fonctionnera pas normalement sans cette ligne

Il faut le rendre exécutable !

En créant un nouveau fichier, ces droits de base sont toujours en écriture et lecture maximum. Il faut penser à faire **chmod +x script.sh** (penser à remplacer « script.sh » par le nom du script !)

chmod : quelques explications

- **Les utilisateurs dans un système UNIX** : sont associés à des groupes. Certains groupes ont des autorisations spécifiques, c'est le cas de « root » par exemple
- **Les permissions d'un fichier** : en faisant **ls -l**, les permissions accordées sur un fichier sont affichées au format rwx-rwx-rwx : Read, Write, eXecute pour le propriétaire (user), les utilisateurs du même groupe (group) et les autres (other). Ces permissions peuvent s'exprimer en lettre ou en binaire (voir slide suivante)
- **Modifier les permissions** : chmod prend deux paramètres : les autorisations à accorder, et le fichier/dossier concerné.

Quelques exemple de chmod

`chmod u+x toto.py` : accorde le droit d'exécuter le fichier toto.py au propriétaire du fichier

`chmod a-w rules` : retire à tous le droit de modifier le fichier rules

`chmod 764 bonjour.txt` :

- Accorde au propriétaire tous les droits (4 + 2 + 1)
- Accorde au groupe le droit d'écriture et de lecture (4 + 2)
- Accorde aux autres le droit de lecture (4)

L'ordre des permissions sous ce format est toujours : propriétaire, groupe, autres. C'est le même ordre lorsque la commande `ls -l` est utilisée.

Quelques explications
sur le format binaire :

Permission	Read	Write	Execute
Bit	bit de gauche	bit du milieu	bit de droite
En base 2	100	010	001
En base 10	4	2	1

Interagir avec les périphériques

- Dans UNIX/Linux, tout est fichier :
 - Au branchement d'un périphérique, un driver s'installe pour l'utiliser. Ce driver est un fichier présent dans **/dev**
On en aura besoin lorsque l'on utilisera l'Arduino avec la Jetson Nano
- Pour afficher la liste des périphériques USB : **lsusb**
- Pour afficher la liste des partitions (zone de mémoire : carte SD...) : **fdisk -l**



Passons à la pratique

