

TP-LINUX: Prise en main de l'outil

Objectif : Mettre en service un nano-ordinateur Raspberry en mode autonome.
Etablir une connexion SSH par le réseau.

PARTIE 1 : PRESENTATION DU SYSTEME

Pour cette manipulation, nous allons utiliser un « nano-ordinateur », c'est-à-dire un ordinateur de très petite taille avec un microprocesseur différent, utilisant une architecture ARM. Ce type d'architecture est adapté aux systèmes embarqués (faible encombrement, faible consommation, modularité).

Info culture :

La société *ARM Ltd*, offre aux fabricants de puces une architecture de micro-processeur à intégrer à des composants plus spécialisés (Processeurs de cartes vidéo, microprocesseurs, ...). Un fabricant assez connu dans le monde des Smartphones : *Qualcomm*. De son côté, *Apple* a également décidé (après avoir quitté Motorola pour Intel) de remplacer son architecture basée sur Intel par son propre processeur basé sur une architecture ARM.

https://fr.wikipedia.org/wiki/Architecture_ARM

<https://www.zdnet.fr/pratique/processeurs-arm-tout-savoir-tout-comprendre-39907353.htm>

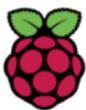
Notre choix se pose sur un RASPBERRY PI (Au labo : PI3 pour l'instant ... Le PI4 est dispo à la vente)

On remarque immédiatement qu'il dispose de toute la connectique d'un ordinateur de type PC : Connecteur HDMI, USB, ETHERNET. Il embarque aussi un module WiFi et BlueTooth.

Mais il offre également un **connecteur pour un module caméra**, et un **connecteur GPIO** pour communiquer avec des capteurs ou systèmes électroniques extérieurs.

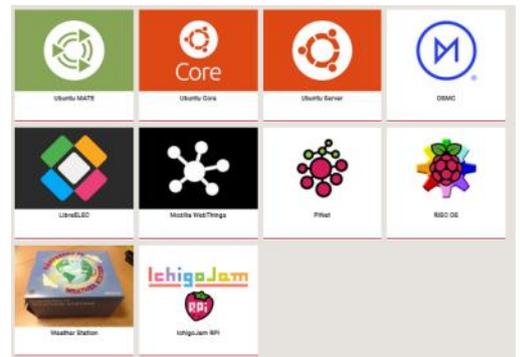


Le **système d'exploitation** est installé sur une carte SD.



Raspberry Pi OS (previously called Raspbian)

Côté système d'exploitation, nous utiliserons *Raspbian* (appelé maintenant *Raspberry Pi OS*), un **Linux** dérivé de la distribution *Debian* (comme *Ubuntu*). Un coup d'œil sur le site officiel (raspberrypi.org) nous apprend qu'il existe d'autres possibilités.



Comme l'objectif est de travailler en mode « embarqué », nous utiliserons le PI sans clavier, ni souris, ni écran... juste par une connexion réseau SSH.

Nous y accéderons ensuite à partir d'un ordinateur du labo.

Sauf pour la phase d'installation !

TRAVAIL : Utilisez la notice d'installation fournie pour rendre votre Pi opérationnel en mode SSH.

PARTIE 2 : TRAVAILLER AVEC LINUX EN MODE SSH (SECURE SHELL)

Pourquoi en SSH ?

Vous avez sans doute l'habitude de travailler dans un environnement graphique (Windows, OS/X, Android, IOS, Linux Desktop Gnome/Kde/ ...).

C'est normal et on souhaite que l'UTILISATION d'un ordinateur reste la plus simple et intuitive possible.

Les choses changent sur des ordinateurs de type SERVEURS, des ROUTEURS, ou dans le cas de la MAINTENANCE des ordinateurs :

- Il n'y a pas forcément une interface graphique installée.
- Un administrateur réseau préférera travailler en mode « console », sans l'environnement graphique, pour accéder plus rapidement à des paramètres systèmes.
- Le mode console permet de créer des scripts pour automatiser certaines tâches.
- Le travail en mode « console » devient même souhaitable lorsque la maintenance se fait à distance et qu'on ne veut pas utiliser des solutions lourdes comme MSTSC (Connexion Bureau à distance), VLC, Anydesk, Team Viewer ...



SSH : c'est un protocole réseau qui permet la connexion sécurisée en mode « console » à un système équipé d'un service « serveur ssh ».

Le client SSH :

Dans votre installation de Raspberry OS, vous avez activé le service « Serveur SSH »

Vous allez maintenant vous connecter à ce service avec un CLIENT SSH à partir d'un poste de travail Windows avec un des 2 logiciels suivants :



BitVise : **Notre préféré** ! Un outil performant qui permet également le transfert de fichiers (par SFTP). Le SFTP autorise aussi de travailler directement sur les fichiers Linux à partir d'un éditeur de texte de Windows (Notepad++¹ par exemple). Cela évite de travailler avec Nano.



Un logiciel compact et rapide facile à mettre en service.

¹ Utiliser un éditeur Windows pour travailler sur un fichier Linux via SFTP ? C'est une bonne idée. Grand confort de travail. Il faut juste penser à modifier la façon dont Notepad++ fait ses fins de ligne. Le symbole n'est pas le même sous Windows et sous Linux. A modifier dans Notepad++ :
menu Edition → Convertir les sauts de ligne → Linux
menu Encodage → UTF8

TRAVAIL :

- **(Pour le Labo SN E203 à E205) :** Dans une console « CMD » de votre poste de travail Windows, tapez la commande :

```
route add 192.168.xx.0 mask 255.255.255.0 10.69.88.1
```

NB : Demander confirmation au professeur pour la valeur de **xx**

Cette commande ajoute à votre PC (qui fait partie du réseau LABO) une **ROUTE TCP/IP** vers le réseau BOX (prises A) sur lequel se trouve le Raspberry.

- Utilisez un des 2 logiciels pour ouvrir une connexion SSH avec votre Raspberry. C'est l'adresse IP qui permet de cibler le bon Raspberry. Vous avez dû la noter lors de l'installation.

Vous devez arriver à un écran de ce type :

**PARTIE 3 : ET ENSUITE ?**

Votre environnement de travail « technicien – administrateur système » est prêt à fonctionner.

Vous pouvez par exemple forcer une mise à jour du système (Option !! Demandez au professeur).

Site intéressant : <https://www.lecoindunet.com/difference-apt-update-upgrade-full-upgrade>

```
sudo apt update           // pour MAJ de la liste des magasins (long)
sudo apt upgrade         // pour la mise à jour des parquets logiciels (très long !)
```

Vous pouvez également changer le mot de passe de votre utilisateur « pi » avec la commande :

```
passwd
```

ATTENTION : Si vous perdez le mot de passe il faudra TOUT réinstaller !!