

Cours Systèmes d'Exploitation Embarqués

Par:

Hassan EL FADIL

elfadil.h@gmail.com

PLAN

CH-1 OS Embarqués et Plateformes

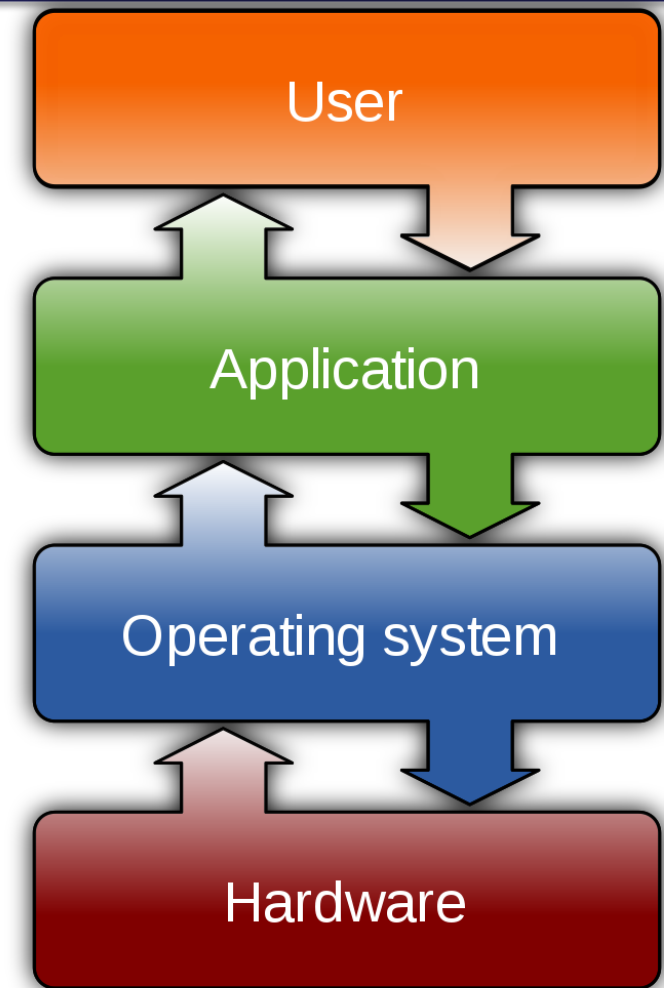
CH-2 Installation de l'OS Raspbian sur Raspberry-Pi

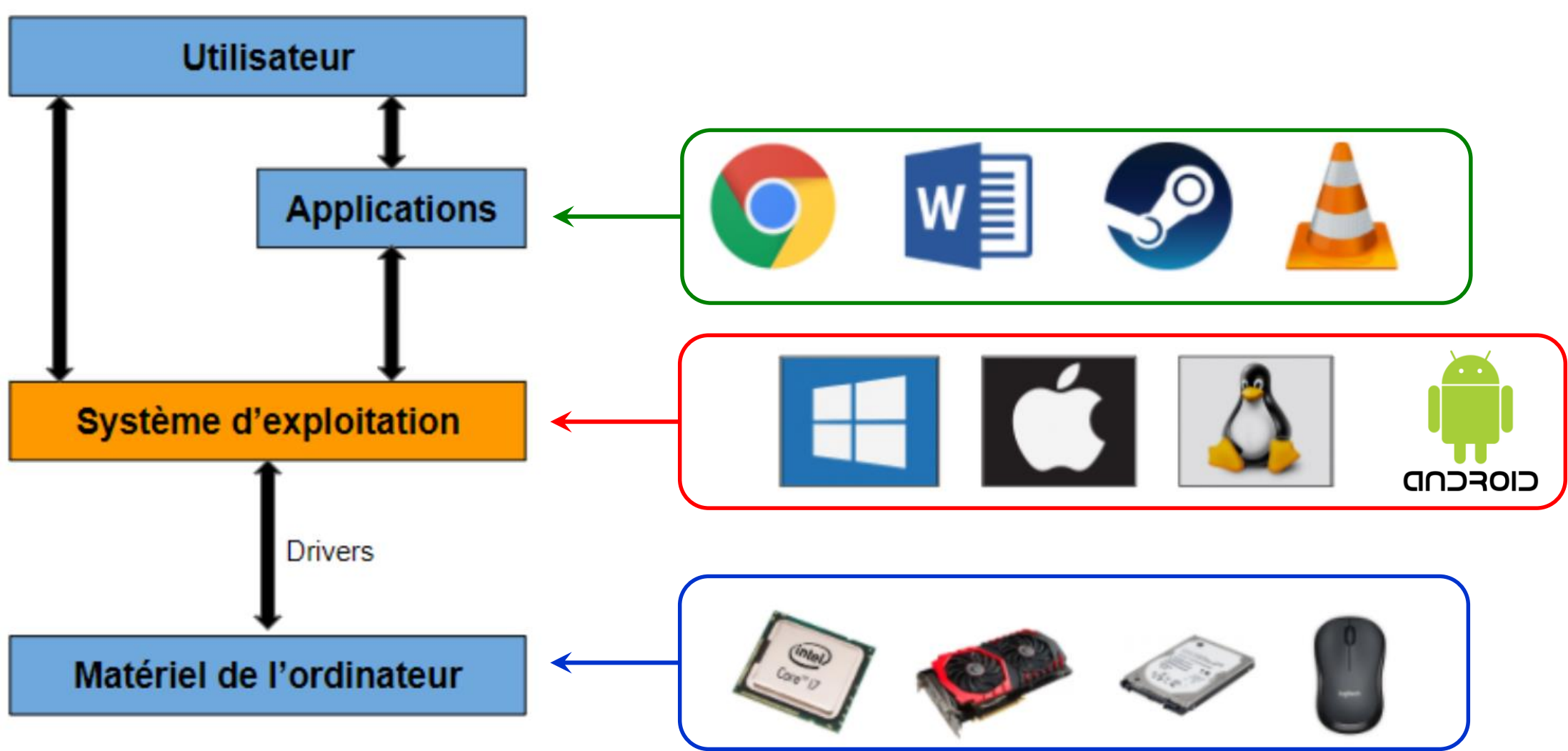
CH-3 Commandes LINUX en utilisant Raspberry-Pi

CH-4 Python sur Raspberry-Pi

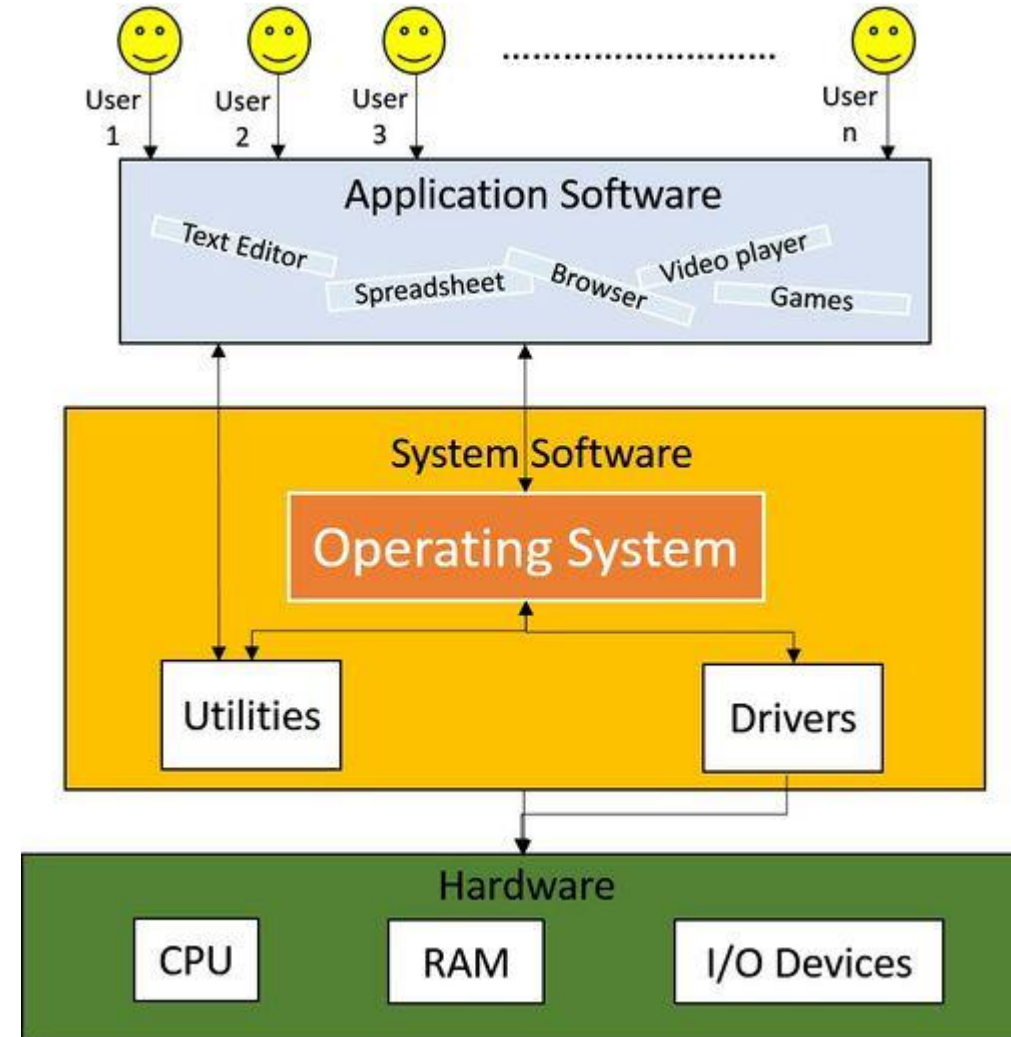
CH1: OS Embarqués et Plateformes

- Un système d'exploitation embarqué (OSE, ou OS pour "**Operating System**" en anglais) est un logiciel conçu pour gérer les fonctions de base d'un système informatique embarqué.
- Les OSE sont des systèmes informatiques **intégrés à d'autres appareils**, tels que:
 - *des appareils électroniques grand public,*
 - *des dispositifs médicaux,*
 - *des systèmes de contrôle industriel,*
 - *des véhicules, etc.*

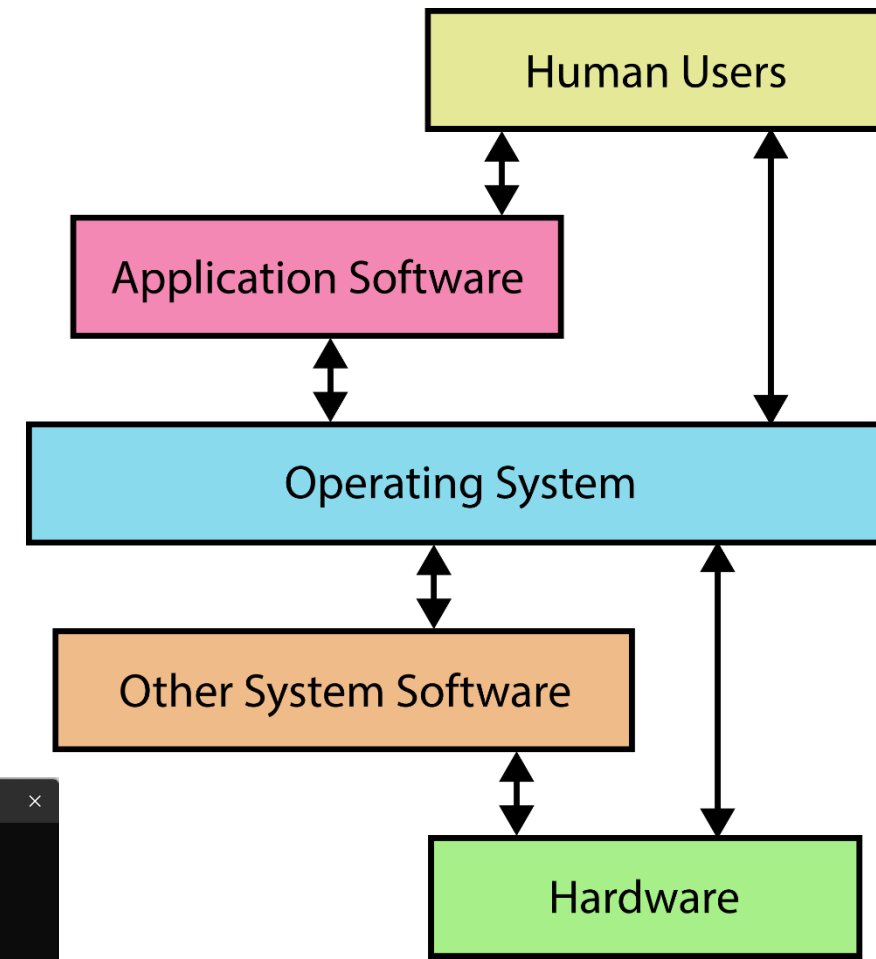
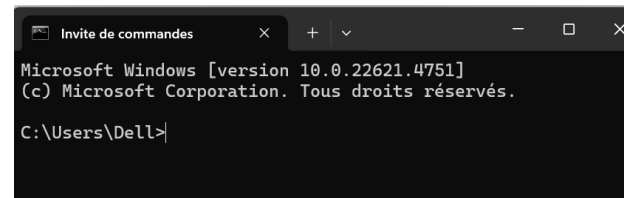




- L'OS est le **premier programme** exécuté au démarrage d'une machine
- Il assure **le lien** entre les ressources matérielles de la machine et les utilisateurs.
- L'OS **affecte des ressources** (stockage, mémoire, calcul) aux applications pour éviter que leur utilisation n'interfère avec d'autres applications.
- Les utilisateurs **interagissent** avec les applications qui, à leur tour, communiquent avec l'OS.



- Néanmoins les utilisateurs peuvent **communiquer** directement avec l'OS (demander des services) moyennant:
 - *une interface graphique (GUI)*
 - *des lignes de commande (CLI)*
 - *ou des interfaces de programmation.*

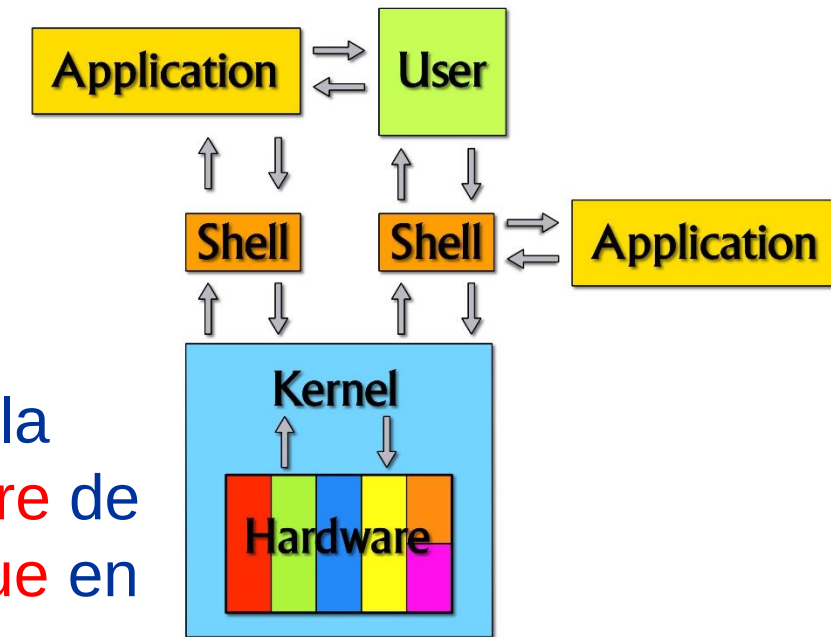


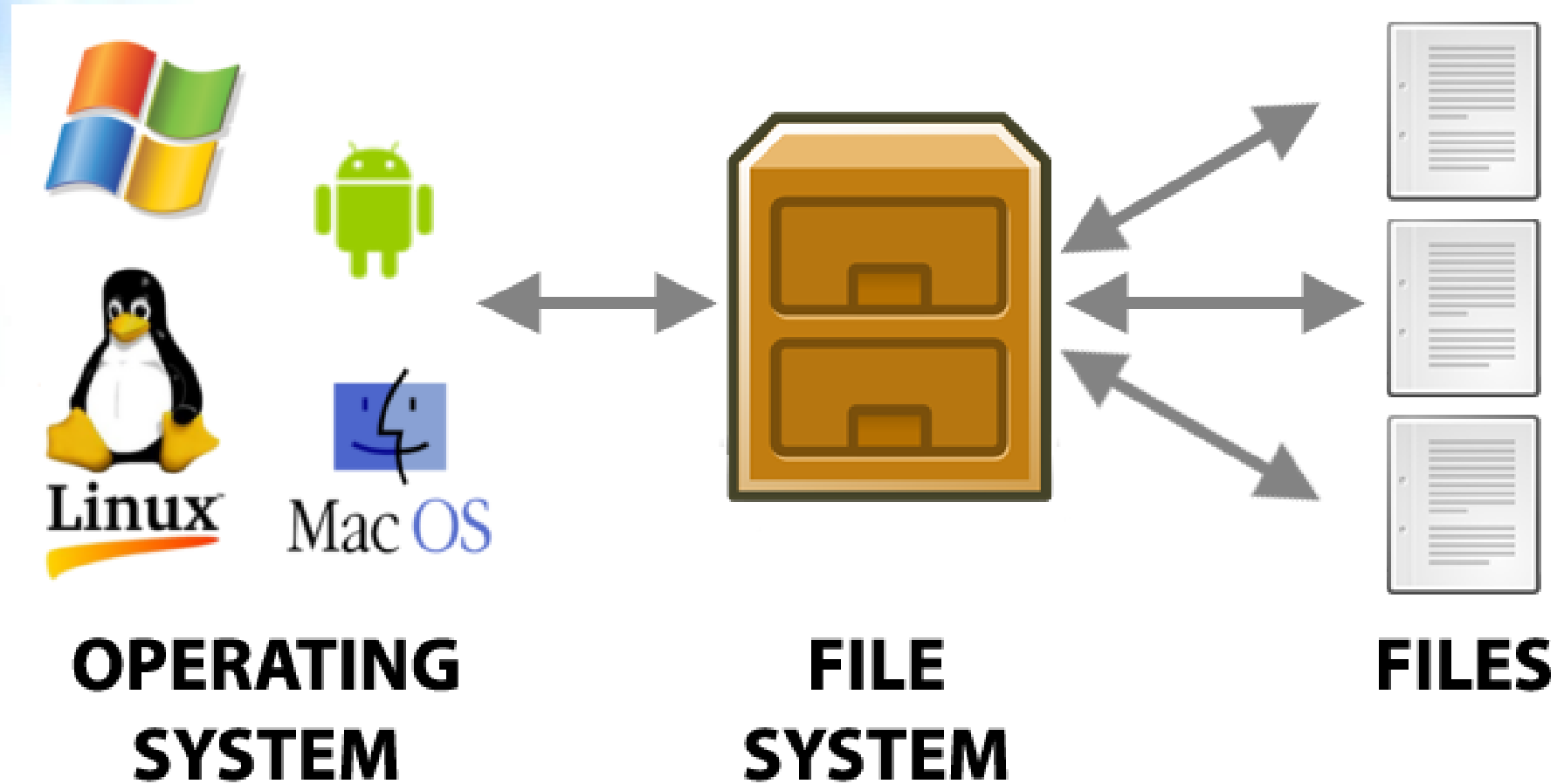
Les fonctions d'un système d'exploitation comprennent :

- Charger **une interface logicielle** permettant à l'utilisateur d'interagir (généralement une interface graphique (**GUI**: Graphical User Interface) ou en ligne de commande (**CLI**: Command-line Interface))
- Gérer **les ressources matérielles** telles que la mémoire et les périphériques
- Gestion des **utilisateurs**
- Authentification et sécurité
- Gestion des **fichiers**

3. Les principaux composants d'un OS

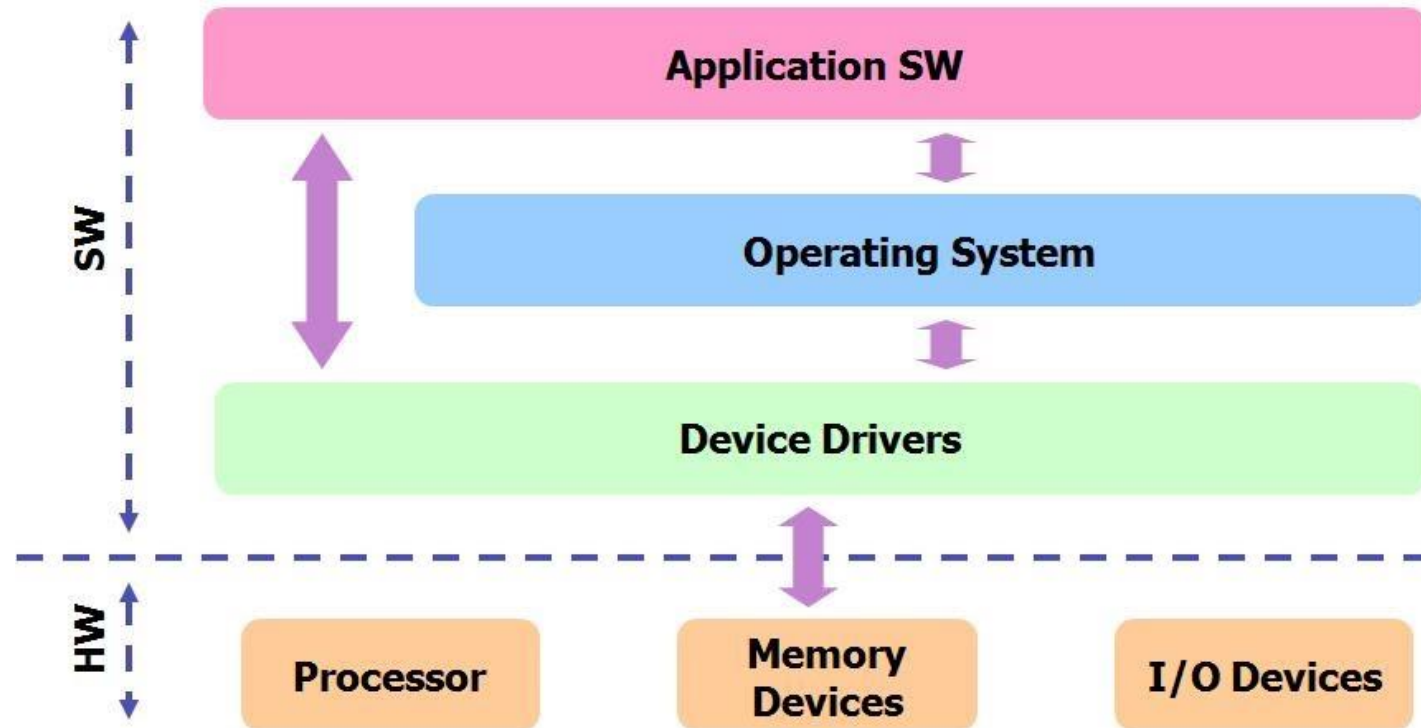
- 1) **KERNEL (Noyau):** Il est responsable de l'exécution des instructions fondamentales nécessaires au fonctionnement du système.
- 2) **SHELL (interpréteur de commande):** permet de communiquer avec l'OS soit par CLI ou par GUI.
- 3) **FILE SYSTEM (Système de fichiers):** gère l'arborescence, l'écriture et la lecture des fichiers. Cela inclut la création, la suppression, la lecture, et l'écriture de fichiers. Il organise également les fichiers sur le disque en une structure logique, facilitant ainsi leur gestion





4) DRIVERS (Pilotes): permettent la gestion de périphériques

C'est un **logiciel intermédiaire** qui permet au système d'exploitation (Windows, Linux, etc.) de **communiquer** avec un matériel (imprimante, carte graphique, clavier, etc.).



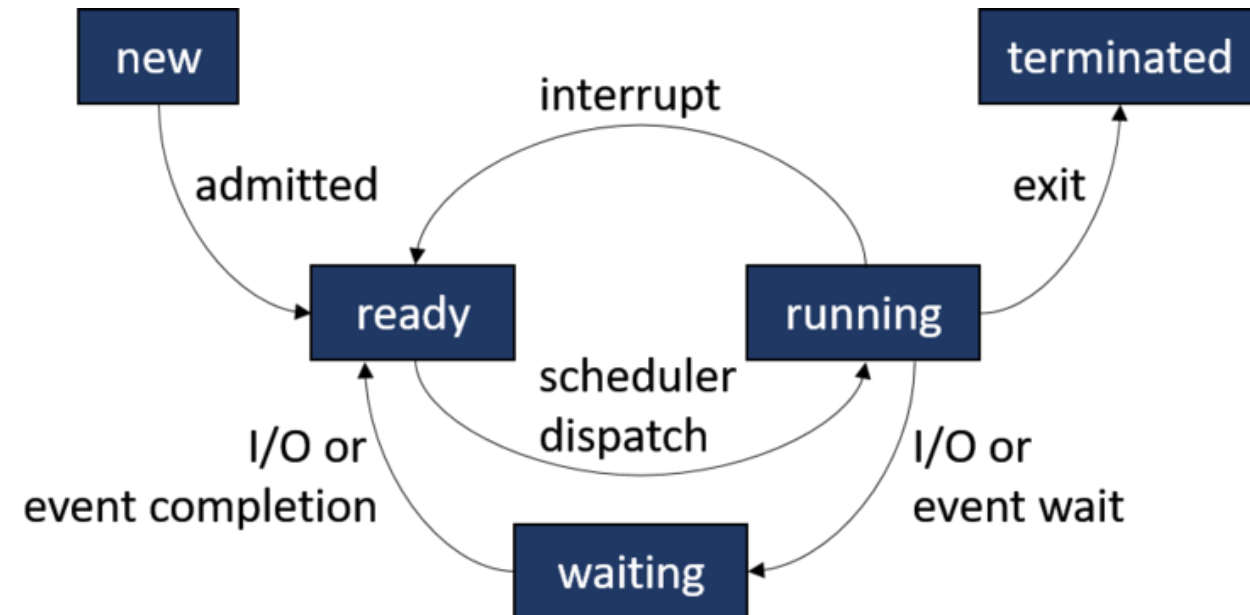
4. Notion de Processus (Process)

- Un **processus** est un **programme** en cours d'exécution, avec son propre espace mémoire, ses ressources système et son état d'exécution.
- Il s'agit de **l'entité fondamentale** à laquelle le système d'exploitation **alloue des ressources** processeur et mémoire.
- Chaque processus a son propre **contexte d'exécution**, y compris sa **pile**, son espace mémoire, ses registres et d'autres attributs.
- Ces attributs permettent au système d'exploitation de gérer l'exécution **simultanée** de **plusieurs processus** tout en les isolant les uns des autres.

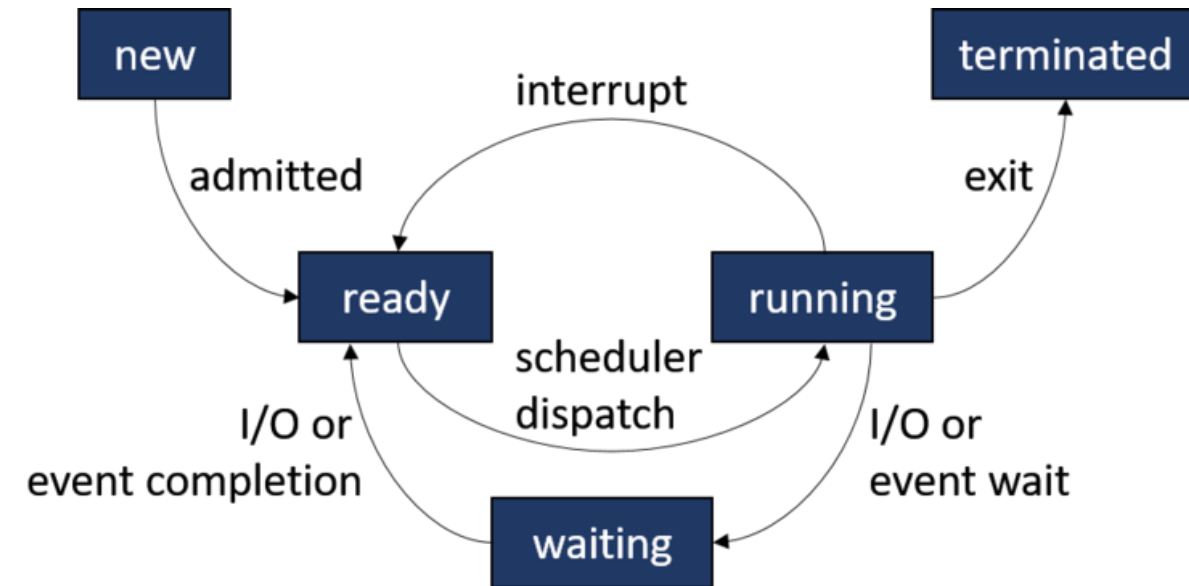
5. Gestion des processus (cycle de vie d'un processus)

Un système d'exploitation peut faire traverser à un processus de nombreux **états possibles** :

- **Nouveau (New)** : Le processus est en cours de création.
- **Prêt (Ready)** : Le processus est prêt à être exécuté, mais le processeur ne lui a pas encore été attribué.



- **En cours d'exécution (Running)** : Le processus est actuellement en cours d'exécution sur le processeur.
- **En attente (Waiting)** : Le processus attend une ressource ou un événement, comme une entrée/sortie.
- **Terminé (Terminated)** : Le processus a terminé son exécution.



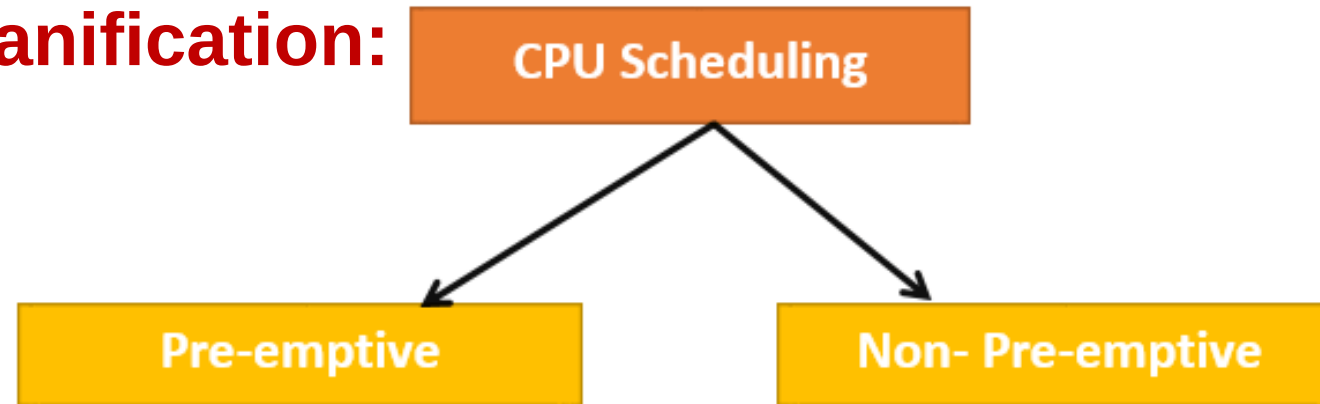
Exemple:
Gestionnaire des
tâches sous Windows

Gestionnaire des tâches					
Fichier Options Affichage					
Processus Performance Historique des applications Démarrage Utilisateurs Détails Services					
Nom	PID	Description	Statut	Groupe	
AarSvc		Agent Activation Runtime	Arrêté	AarSvcGroup	
AarSvc_efe6a9f		Agent Activation Runtime_efe...	Arrêté	AarSvcGroup	
AdobeARMservi...	17000	Adobe Acrobat Update Service	En cours d'exéc...		
AJRouter		Service de routeur AllJoyn	Arrêté	LocalServiceNet...	
ALG		Service de la passerelle de la c...	Arrêté		
AppIDSvc		Identité de l'application	Arrêté	LocalServiceNet...	
Appinfo	4424	Informations d'application	En cours d'exéc...	netsvcs	
AppMgmt		Gestion d'applications	Arrêté	netsvcs	
AppReadiness		Préparation des applications	Arrêté	AppReadiness	
AppVClient		Microsoft App-V Client	Arrêté		
AppXSvc	9732	Service de déploiement AppX...	En cours d'exéc...	wsappx	
AssignedAccess...		Service AssignedAccessMana...	Arrêté	AssignedAcces...	
AudioEndpointB...	2012	Générateur de points de termi...	En cours d'exéc...	LocalSystemNe...	
Audiosrv	3852	Audio Windows	En cours d'exéc...	LocalServiceNet...	

6. Planification des processus (Process Scheduling)

- Un processus attend des **instructions du CPU** pour passer de l'état prêt à l'état en cours d'exécution.
- Il incombe l'OS d'analyser et de déterminer quels processus doivent **passer à l'état RUNNING** et lesquels doivent être maintenus à l'état **READY**.
- Cette fonction de l'OS est appelée la **planification** (ou l'**ordonnancement** des processus (**Process Scheduling**)).
- L'OS tente de **maintenir le CPU occupé** et d'exécuter des processus tout en **garantissant** à chaque programme le temps de réponse **le plus court possible**.

Catégories de planification:

**Préemptif :**

- Le temps CPU est divisé en créneaux temporels et les processus se voient attribuer un créneau pour passer à l'état « **Running** » et s'exécuter.
- Si un processus ne parvient pas à terminer son exécution dans le créneau temporel attribué, il est déplacé à l'état « **Ready** » jusqu'à ce qu'il soit rappelé.

Non Préemptif:

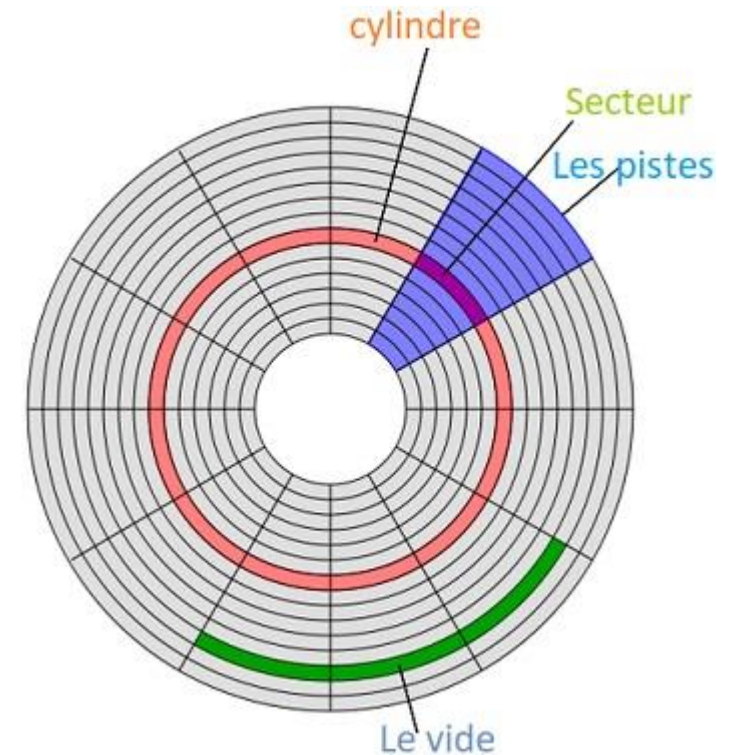
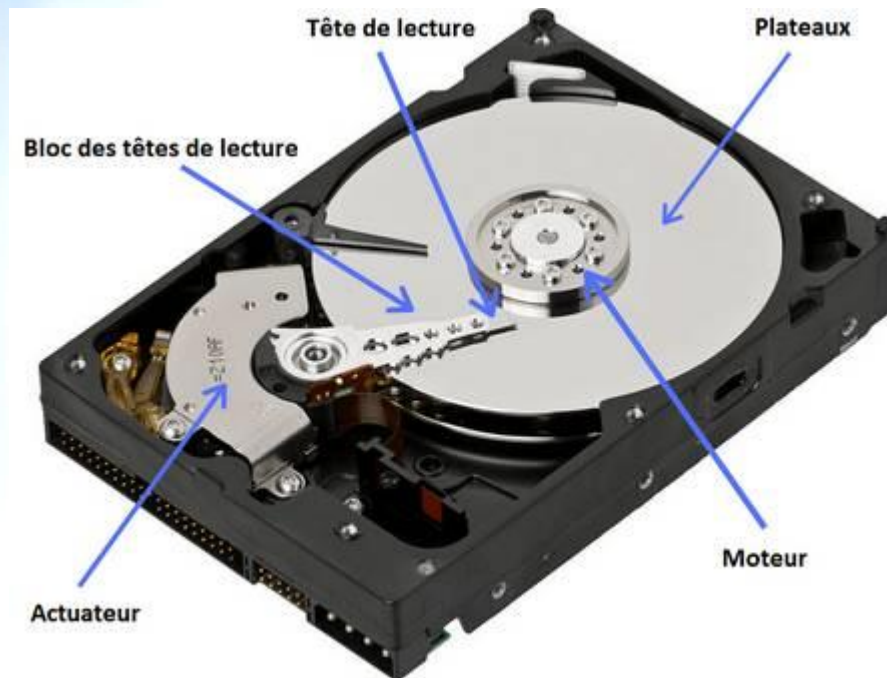
Un processus est maintenu à l'état « **Running** » jusqu'à ce qu'il termine son exécution ou passe à un état « **Waiting** » (par exemple, en attendant la fin d'un événement d'E/S).

Dans le cas de ce dernier, un autre processus aura l'opportunité de s'exécuter.

7. Systèmes de fichiers

- **Un système de fichiers:** est une **méthode** utilisée par les OS pour **organiser et stocker** des données sur des **dispositifs de stockage**, tels que des **disques durs**, des **clés USB**, des **cartes SD**, et d'autres supports de stockage.
- Il définit la manière dont les fichiers sont nommés, organisés, et stockés sur le support physique.
- **Secteur:** Les données sont stockées en utilisant des secteurs et des clusters. Un secteur est la plus petite unité adressable en lecture/écriture (**Capacité minimum de 4ko**).
- **Un cluster:** est un ensemble de secteurs. La **taille d'un cluster** dépend du système de fichiers utilisé.

Anciens Disques durs (HDD – Hard Disk Drives)



Disques SSD (Solid State Drives)

Ce sont des disques à **mémoire flash**, donc sans pièces mobiles.



HDD



SSD



FAT (File Allocation Table) : FAT32

NTFS (New Technology File System) : Utilisé principalement sur les systèmes d'exploitation Windows.

exFAT (Extended File Allocation Table) : Conçu pour être utilisé sur des supports amovibles tels que les cartes SD et les clés USB. Il prend en charge des fichiers de taille supérieure à FAT32 et fonctionne bien avec des volumes importants.

APFS (Apple File System) : Utilisé par défaut sur les systèmes d'exploitation macOS depuis macOS High Sierra.

ext4 (Fourth Extended File System) : Un système de fichiers couramment utilisé sur les distributions Linux. Il offre des améliorations par rapport à son prédécesseur, ext3, notamment en termes de capacité de stockage et de performances.

8. “Le Chargeur: Loader”

- Les programmes et bibliothèques doivent être chargés dans le système d'exploitation avant de pouvoir être exécutés.
- Ceci est effectué par un composant du système d'exploitation appelé chargeur (Loader).
- Le flux habituel d'un chargeur est le suivant :
 - Extraire les instructions du programme/bibliothèque de son fichier exécutable dans la mémoire secondaire vers la RAM.
 - Effectuez toutes les tâches supplémentaires requises pour préparer l'exécution du programme/de la bibliothèque.
 - Passez le contrôle au programme/à la bibliothèque pour exécuter ses instructions.

9. Notion de multitâche et de multiprogrammation

- La "**notion de multitâche**" fait référence à la capacité d'exécuter plusieurs tâches ou processus en même temps.
- Un système d'exploitation multitâche permet à plusieurs programmes de s'exécuter **simultanément**, même si le processeur du système ne peut exécuter qu'une seule instruction à la fois.
- Il existe deux principaux types de multitâche :
 - 1.Préemptif** : Le système d'exploitation peut interrompre l'exécution d'une tâche pour donner la possibilité à une autre de s'exécuter. Cela se fait généralement en **fonction de priorités** attribuées aux tâches.
 - 2.Coopératif** : Les tâches ou processus **coopèrent** pour partager le temps processeur. Chaque tâche doit **volontairement** libérer le contrôle pour permettre à une autre tâche de s'exécuter.

les utilisateurs interagissent avec les systèmes d'exploitation en utilisant plusieurs méthodes :

1. ligne de commande : Command Line Interface (CLI):

- Il s'agit d'une **interface textuelle** dans laquelle l'utilisateur peut donner des **commandes** à l'aide d'une méthode de saisie, telle qu'un clavier, pour commander au système d'exploitation d'exécuter des processus et des scripts.

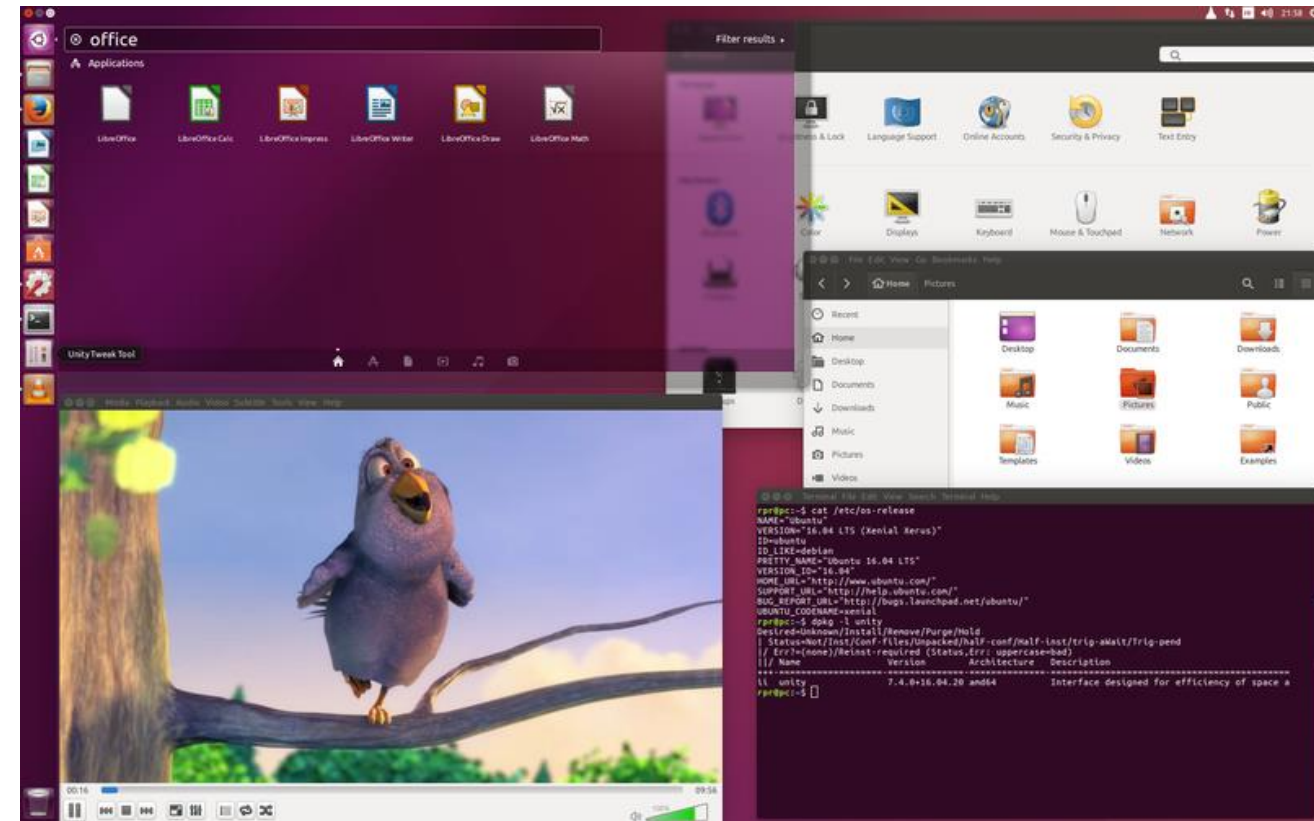
Exemples :

- Invite de commande (cmd)** de Windows
- Shell **bash** sous Linux

```
root ~ # ping google.com
PING google.com (74.125.95.103) 56(84) bytes of data.
64 bytes from iw-in-f103.1e100.net (74.125.95.103): icmp_seq=1 ttl=47 time=15.3
ms
^C
--- google.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 15.453/15.453/15.453/0.000 ms
root ~ # ls
Desktop  README
root ~ # cd /
root / # ls
bin      dev      home    lost+found  mnt      proc      sbin      srv      tmp      var
boot     etc      lib      media       opt      root      shadow    sys      usr
root / # pacman -Ss pidgin
extra/libpurple 2.6.6-1
  IM library extracted from Pidgin
extra/pidgin 2.6.6-1
  Multi-protocol instant messaging client
extra/pidgin-encryption 3.0-3
  A Pidgin plugin providing transparent RSA encryption using NSS
extra/purple-plugin-pack 2.6.3-1
  Plugin pack for Pidgin
extra/telepathy-haze 0.3.4-1 (telepathy)
  A telepathy-backend to use libpurple (Pidgin) protocols.
community/guifications 2.16-1
  A set of GUI popup notifications for pidgin
community/pidgin-fonemobutton 0.1.6-1
  Adds a video-chat button to the the conversation window
community/pidgin-libnotify 0.14-3
  pidgin plugin that enables popups when someone logs in or messages you.
community/pidgin-musictracker 0.4.21-2
  A plugin for Pidgin which displays the music track currently playing.
community/pidgin-otr 3.2.0-1
  Off-the-Record Messaging plugin for Pidgin
root / #
```

2. Interface graphique : Graphical User Interface (GUI)

- L'utilisateur se voit présenter une interface contenant des fenêtres, des icônes, des menus, etc.
- Une interface graphique offre une interaction plus conviviale avec le système d'exploitation



...Interaction avec l'utilisateur

Avantages de "CLI"

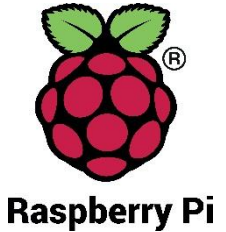
Voici quelques **avantages** spécifiques de la **CLI** par rapport à la **GUI** :

- 1.Efficacité pour les tâches répétitives** : Les utilisateurs expérimentés peuvent créer des scripts ou des commandes qui automatisent des séquences d'actions, ce qui est plus rapide que de naviguer à travers une interface graphique.
- 2.Utilisation des ressources système** : La CLI utilise généralement moins de ressources système comparée à une GUI.
- 3.Contrôle précis** : Les utilisateurs avancés apprécient souvent la CLI pour son contrôle précis sur les opérations.
- 4.Scripting et automatisation** : La CLI facilite l'automatisation des tâches grâce à la création de scripts. Les utilisateurs peuvent créer des scripts pour exécuter une série d'instructions, ce qui est utile pour les tâches complexes ou répétitives.
- 5.Utilisation à distance** : La CLI est souvent préférée pour l'administration à distance des systèmes, car elle peut être utilisée via des connexions réseau sans avoir besoin d'une interface graphique. Cela peut être crucial pour la gestion des serveurs.
- 6.Apprentissage approfondi du système** : Les utilisateurs de la CLI ont souvent une compréhension plus approfondie du système. En utilisant des commandes spécifiques, ils peuvent obtenir des informations détaillées sur le fonctionnement interne du système.

11. Systèmes d'exploitation courants

Élément de Comparaison	Windows	MacOs	Linux
Matériel	Compatibilité avec une large gamme de périphériques	Bien que MacOs autorise également l'utilisation de pilotes tiers, de nombreux appareils choisissent de fournir uniquement des pilotes officiels pour Windows	Nombreux fabricants de périphériques développement plus de pilotes pour Windows et MacOs au lieu de Linux.
Logiciel	Offre la plus haute compatibilité avec les logiciels du marché.	Par rapport à Windows, MacOs a tendance à avoir une plus petite collection de logiciels	Linux a moins d'applications développées pour lui.
Sécurité	Windows dispose d'un logiciel antivirus intégré nommé Windows Defender et de nombreux autres logiciels antivirus peuvent être utilisés.	Plus sûre que Windows. Dispose d'un antivirus intégré nommé Gatekeeper.	Linux est souvent considéré comme le système d'exploitation le plus sûr des trois plates-formes.

Raspberry Pi : Le Raspberry Pi est une carte très populaire qui supporte divers systèmes d'exploitation embarqués tels que **Raspbian** (basé sur Linux), **Windows 10 IoT Core**, et d'autres.



NVIDIA Jetson : Ces cartes, telles que la NVIDIA Jetson Nano, sont conçues pour les **applications d'intelligence artificielle** et prennent en charge des systèmes d'exploitation tels que Ubuntu.



BeagleBone : La BeagleBone est une autre carte embarquée qui prend en charge divers systèmes d'exploitation tels que **Debian**, **Ubuntu**, **Android**, et d'autres.



Orange Pi : Les cartes Orange Pi prennent en charge divers systèmes d'exploitation, y compris **Android**, **Debian**, **Ubuntu**, et d'autres.



Odroid : Les cartes Odroid, comme l'**Odroid-C2** et l'**Odroid-XU4**, prennent en charge divers systèmes d'exploitation tels qu'**Ubuntu**, **Android**, et d'autres.



Pine64 : Les cartes **Pine64**, comme le **Pine A64**, peuvent exécuter différents systèmes d'exploitation tels que **Debian**, **Ubuntu**, et d'autres.



FriendlyARM : Les cartes de FriendlyARM, telles que le **NanoPi**, sont compatibles avec divers systèmes d'exploitation, y compris **Linux**.



Arduino : Bien que les cartes Arduino ne soient pas traditionnellement utilisées pour exécuter des systèmes d'exploitation complets, certaines variantes plus puissantes comme l'**Arduino Yún** peuvent exécuter des distributions Linux légères (**OpenWrt**).



La Raspberry Pi:

Le nom "Raspberry Pi" vient de deux sources principales:

- Le terme "Raspberry" a été choisi par les créateurs du projet en référence aux traditions britanniques de donner des noms de **fruits** aux **ordinateurs**, comme c'était le cas avec l'ordinateur **Acorn** (**Gland**).
- Le mot "**Pi**" fait référence à **Python**, le langage de programmation pour lequel le Raspberry Pi est souvent utilisé.
- L'idée était de créer un nano ordinateur abordable qui encouragerait l'apprentissage de la programmation, en particulier avec le langage Python.

Raspberry : Framboise



Python



Raspberry Pi 4 (Juin 2019)



- SoC – Broadcom BCM2711 **quad-core Cortex-A72** (ARMv8) @ **1.5GHz** with VideoCore VI GPU supporting OpenGL ES 3.0 graphics
- System Memory – 1GB, 2GB or **4GB** LPDDR4 (Low Power Double Data Rate)
- Video Output & Display I/F: 2x micro HDMI ports up to **4Kp60** (résolutions d'écran 4K à une fréquence d'images de 60 Hz)

Raspberry Pi 5 (Octobre 2023)

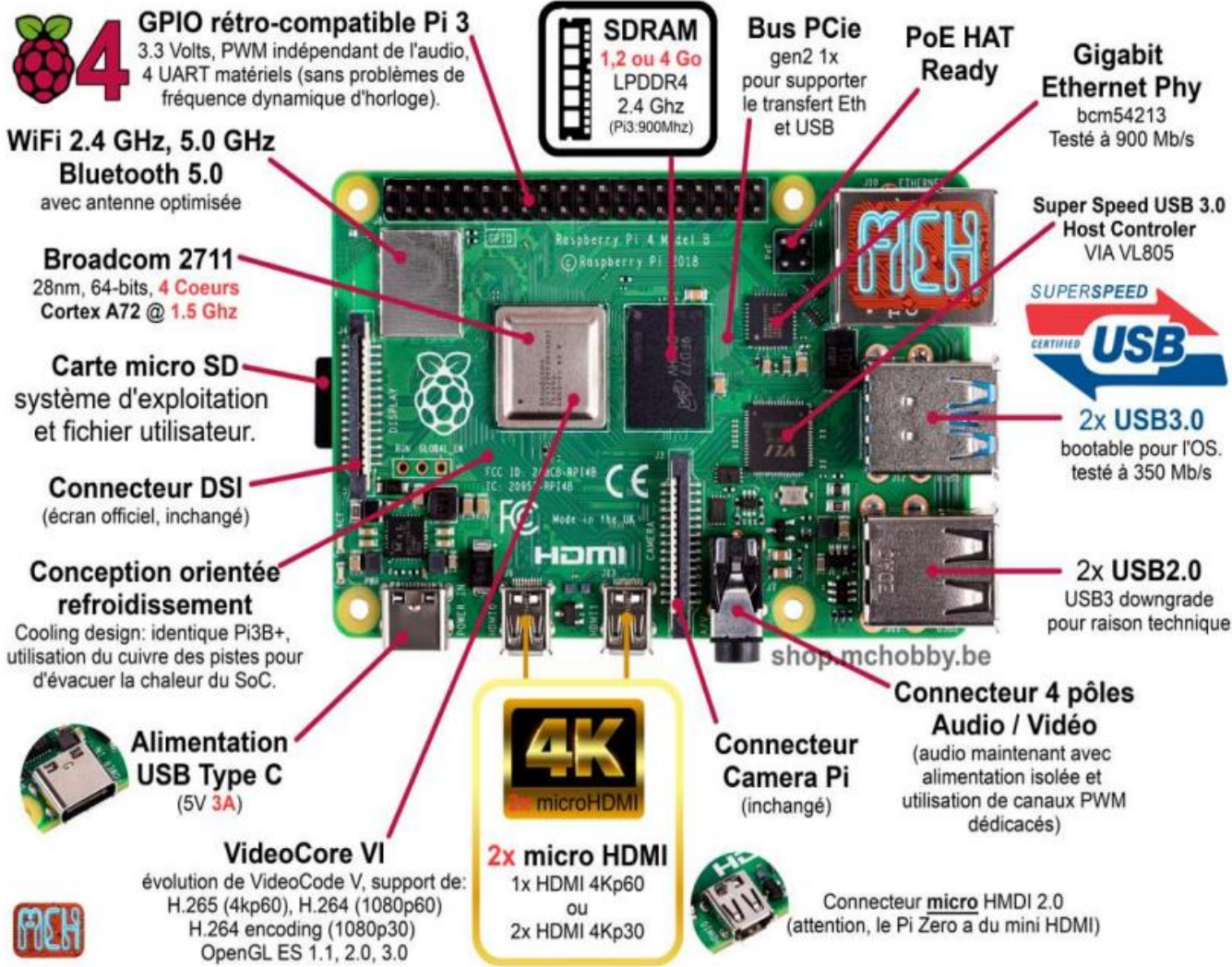


Raspberry Pi 5 features the Broadcom BCM2712 quad-core **Arm Cortex A76** processor @ **2.4GHz**, making it up to three times faster than the previous generation. With RAM variants up to **8GB**.
Nouveau contrôleur **I/O RP1** fournissant diverses interfaces
Composant **DA9091** pour la gestion de l'alimentation.
Connecteurs **FPC** (Flexible Printed Circuit) pour caméra et affichage

1. Présentation de La Raspberry Pi 4

Connecteur **DSI** (Display Serial Interface) est une interface utilisée pour connecter des écrans LCD

PoE HAT ready : signifie que la carte est prête à recevoir un HAT (Hardware Attached on Top) Power over Ethernet (PoE). Cela permet d'alimenter le Raspberry Pi via un **câble Ethernet**, ce qui est particulièrement utile pour des installations où une alimentation électrique traditionnelle n'est pas pratique ou possible



2. Constitution de la carte Raspberry Pi 4

- CPU ARM (cortex A72) quad
- LAN sans fil 802.11n
- Bluetooth 5.0
- Bluetooth faible consommation d'énergie (BLE)
- 1-2-4-8-12 Go de RAM
- 4 ports USB
- 40 broches GPIO
- 2 Ports HDMI (micro)
- Port Ethernet
- Prise jack 3,5 mm combinée
- Interface de l'appareil photo (CSI)
- Interface d'affichage (DSI)
- Emplacement carte micro SD
- VideoCore IV cœur graphique 3D

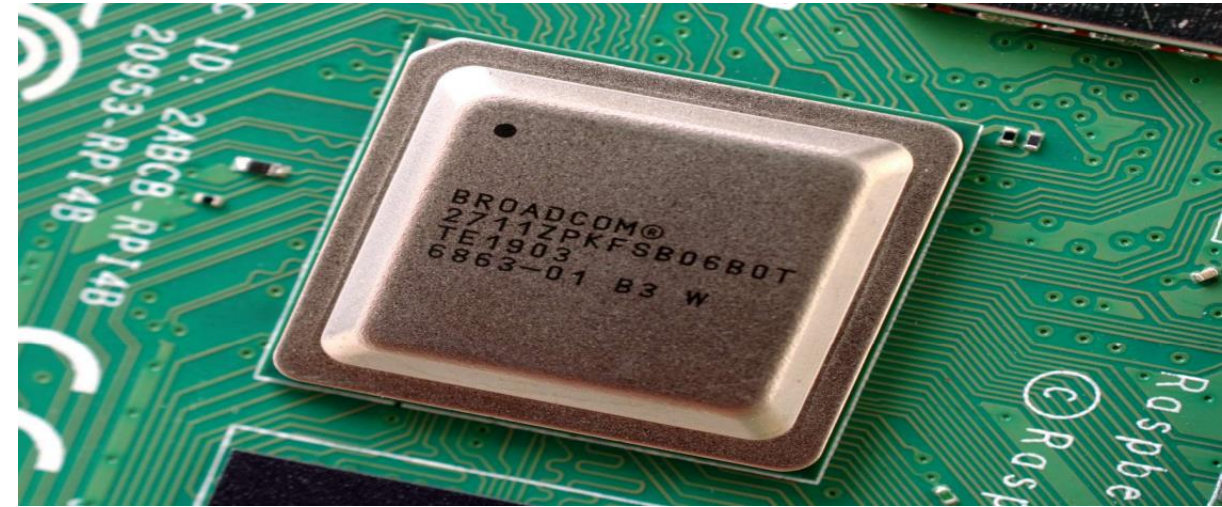
3. Rôle des Périphériques de Raspberry Pi 4

Périphériques	Rôle
Périphériques GPIO (General Purpose Input/Output)	Permettent la communication de la carte avec le monde extérieur
Port Jack	Utilisé pour la sortie audio ou vidéo
USB (Universal Serial Bus)	Permet de connecter souris, cam, ...
Port Ethernet	Utilisé pour la connexion internet
CSI (Camera Serial Interface)	Connexion de caméra
HDMI (High Definition Multimedia Interface)	Connexion d'un écran(TV, vidéoprojecteur)
DSI (Display Serial Interface)	Connexion d'un écran tactile

4. Les principales composantes de Raspberry Pi 4

Système sur puce: Conçu spécialement pour le Pi 4, le système sur puce Braodcom BCM2711 (Soc) comprend quatre processeurs ARM Cortex-A72 haute performance fonctionnant à 1,5 GHz et un processeur graphique VideoCore IV

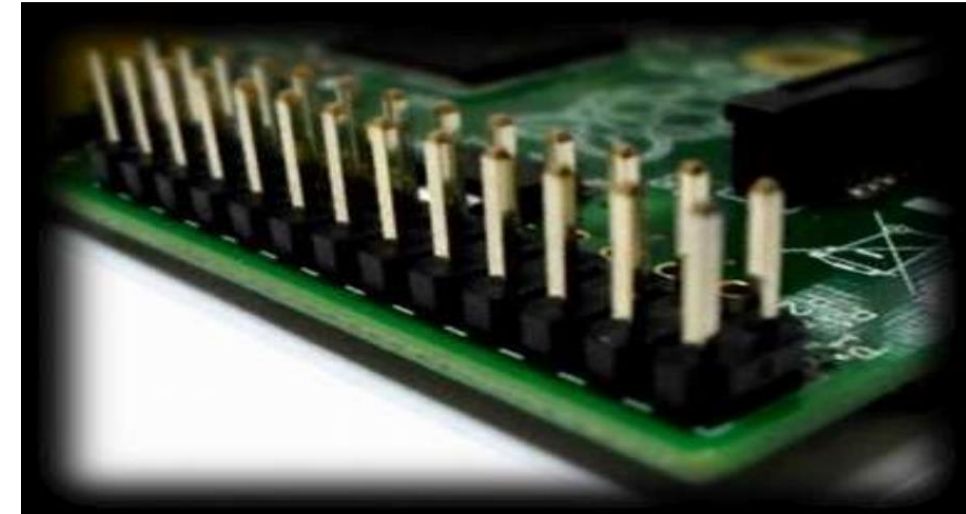
RAM: D'un point de vue pratique, le processeur restant ne devrait pas particulièrement augmenter la puissance brute du Raspberry Pi. En revanche, l'augmentation de la RAM devrait permettre aux utilisateurs ayant une forte consommation de mémoire vive de faire tourner plus d'applications en même temps.



4. Les principales composantes de Raspberry Pi 4

GPIO : La Raspberry Pi 3 ou 4 dispose de 40 broches GPIO

Module sans fil : La puce Broadcom BCM2711 fournit le réseau local sans fil 802.11n de 2,4 GHz, Bluetooth LowEnergy et Bluetooth 5.0



5. Broches GPIO de la Raspberry Pi 4



		Physical Pins					
Function	BCM	pin#	pin#	BCM	Function		
3.3 Volts		1	2		5 Volts		
GPIO/SDA1 (I2C)	2	3	4		5 Volts		
GPIO/SCL1 (I2C)	3	5	6		GND		
GPIO/GCLK	4	7	8	14	TX UART/GPIO		
GND		9	10	15	RX UART/GPIO		
GPIO	17	11	12	18	GPIO		
GPIO	27	13	14		GND		
GPIO	22	15	16	23	GPIO		
3.3 Volts		17	18	24	GPIO		
MOSI (SPI)	10	19	20		GND		
MISO(SPI)	9	21	22	25	GPIO		
SCLK(SPI)	11	23	24	8	CEO_N (SPI)		
GND		25	26	7	CE1_N (SPI)		
RESERVED		27	28		RESERVED		
GPIO	5	29	30		GND		
GPIO	6	31	32	12	GPIO		
GPIO	13	33	34		GND		
GPIO	19	35	36	16	GPIO		
GPIO	26	37	38	20	GPIO		
GND		39	40	21	GPIO		

J8

Power	+3,3V	1	2	+5V	Power
{I2C} SDA1	GPIO2	3	4	+5V	Power
{I2C} SCL1	GPIO3	5	6	GND	
GPCLK0	GPIO4	7	8	GPIO14	{UART} TXD0
GND		9	10	GPIO15	{UART} RXD0
GPIO17		11	12	GPIO18	PCM_CLK
GPIO27		13	14	GND	
GPIO22		15	16	GPIO23	
Power	+3,3V	17	18	GPIO24	
SPI0_MOSI	GPIO10	19	20	GND	
SPI0_MISO	GPIO9	21	22	GPIO25	
SPI0_SCLK	GPIO11	23	24	GPIO8	SPI0_CE0_N
GND		25	26	GPIO7	SPI0_CE1_N
{ID EEPROM}	ID_SD	27	28	ID_SC	{ID EEPROM}
GPCLK1	GPIO5	29	30	GND	
GPCLK2	GPIO6	31	32	GPIO12	PWM0
PWM1	GPIO13	33	34	GND	
PCM_FS	GPIO19	35	36	GPIO16	
GPIO26		37	38	GPIO20	PCM_DIN
GND		39	40	GPIO21	PCM_DOUT

6. Numérotations pour les broches GPIO:

Sur le Raspberry Pi, il existe plusieurs systèmes deux principales numérotations pour les broches GPIO:

- a) **Numérotation BCM (Broadcom SOC channel):** C'est la numérotation basée sur les identifiants de broches du processeur Broadcom utilisé dans le Raspberry Pi.
- b) **Numérotation BOARD:** C'est la numérotation physique des broches sur le connecteur GPIO du Raspberry Pi. Les numéros correspondent à la disposition physique des broches sur la carte.

Il est important de noter que la numérotation **BCM** est souvent utilisée dans les scripts et le code Python pour le contrôle des GPIO, tandis que la numérotation **BOARD** est plus intuitive pour identifier les broches directement sur le Raspberry Pi.

Exemple: GPIO 26

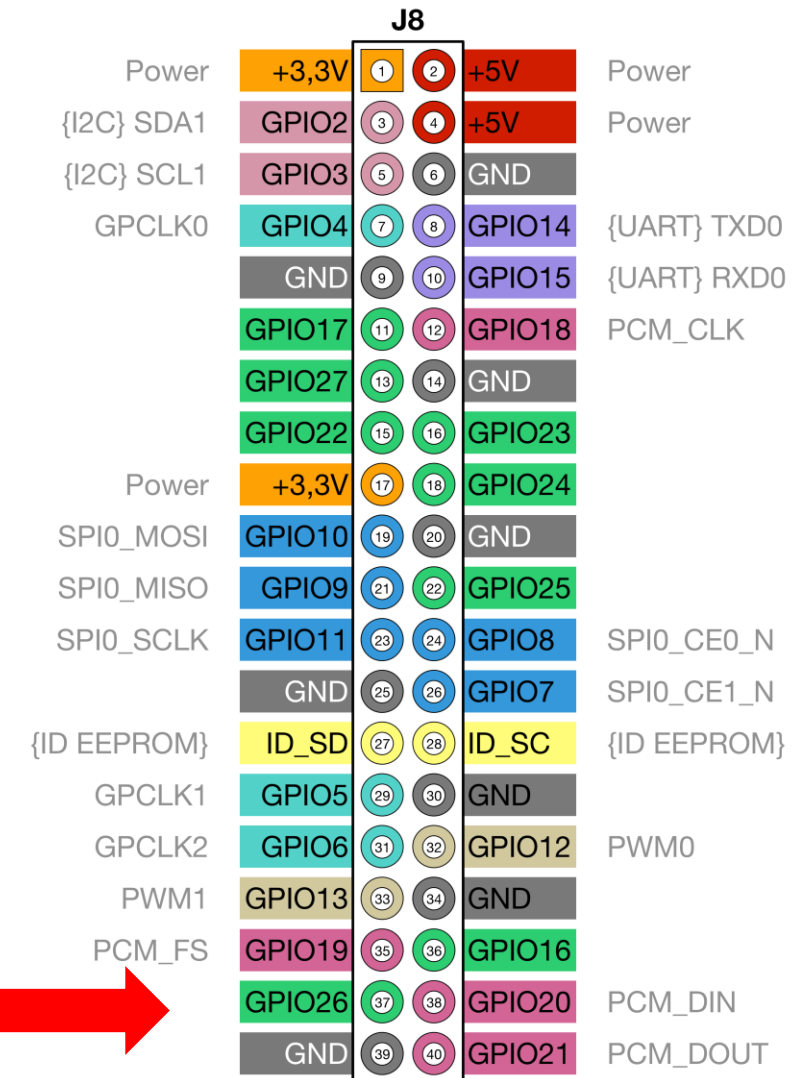
- Numéro BCM (Broadcom) : 26
- Numéro BOARD : 37

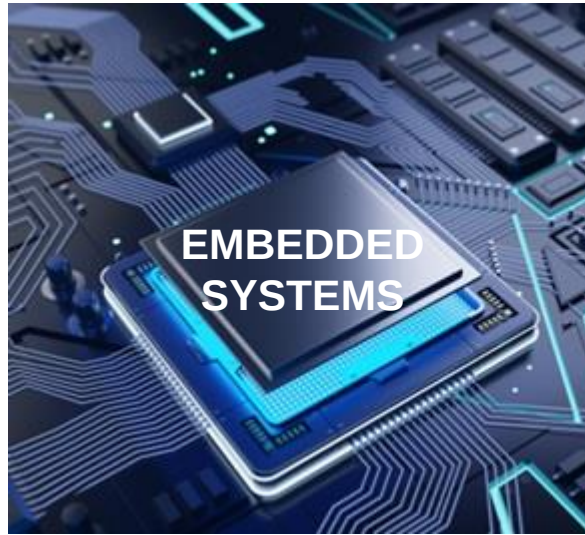
Exemple code Python:

```
import RPi.GPIO as GPIO
import time
# Configuration de la bibliothèque GPIO avec la numérotation BCM
GPIO.setmode(GPIO.BCM)
# Utilisation de la broche GPIO26 (BCM 26)
pin = 26
# Configurer la broche en mode de sortie
GPIO.setup(pin, GPIO.OUT)
# Mettre la broche à l'état haut pendant 2 secondes
GPIO.output(pin, GPIO.HIGH)
time.sleep(2)
# Mettre la broche à l'état bas
GPIO.output(pin, GPIO.LOW)
# Libérer les ressources GPIO
GPIO.cleanup()
```

17/05/2025

H. El Fadil.





FIN !
Questions ?