

Cours Composition d'un PC

Ecrit par Youenn DUVAL

Mail: youenn@barbed.fr

Linkedin: [Youenn DUVAL](#)

Dernière mise à jour : **21/10/2024**

Table des matières

1. [La configuration type d'un ordinateur](#)
2. [Les différents type d'ordinateur](#)
3. [Les Principes de fonctionnement](#)
 1. [Le traitement : le fonctionnement du processeur](#)
 1. [C'est quoi une machine de turing?](#)
 2. [C'est quoi un processeur ?](#)
 3. [Les éléments d'un processeurs](#)
 4. [Comment ça marche ?](#)
 5. [C'est quoi des instructions? ou jeux d'instructions.](#)
 2. [Les autres composants d'un ordinateur](#)
 3. [Le stockage : mémoires de masse, vive, morte, virtuelle, non volatile](#)
 4. [Les périphériques](#)
 5. [Mais comment tout ça communique ?](#)
 1. [Le binaire, les bits et les octets](#)
 1. [Compter en binaire](#)
 2. [En informatique](#)
 3. [8bits, 16 bits, 32bits, 64 bits ...](#)
 2. [test](#)
 6. [Le bios](#)
 7. [Secure Boot](#)
 8. [Les Drivers et Firmware](#)
 9. [Les disques](#)
 1. [Quelques informations supplémentaire sur les partitions](#)
 1. [MBR \(Master Boot Record\)](#)
 2. [GPT \(GUID Partition Table\)](#)
 3. [Différences clés :](#)
 2. [test](#)
 10. [Le système d'exploitation \(OS\)](#)
 1. [Exemple d'OS](#)
 2. [Composition d'un OS](#)
 1. [Noyau \(Kernel\)](#)
 2. [Gestion des processus](#)
 3. [Gestion de la mémoire](#)
 4. [Différence entre Programme et Processus](#)
 3. [Système de fichiers \(File System\)](#)
 4. [Pilotes de périphériques \(Device Drivers\)](#)
 5. [Interface utilisateur](#)
 6. [Gestion des entrées/sorties \(I/O Management\)](#)
 7. [Services systèmes et utilitaires](#)
 8. [Les processus](#)
 9. [Les API](#)
 1. [Les API Web](#)
 2. [Les API Windows](#)

10. [Ordonnancement des processus](#)

11. [POSIX](#)

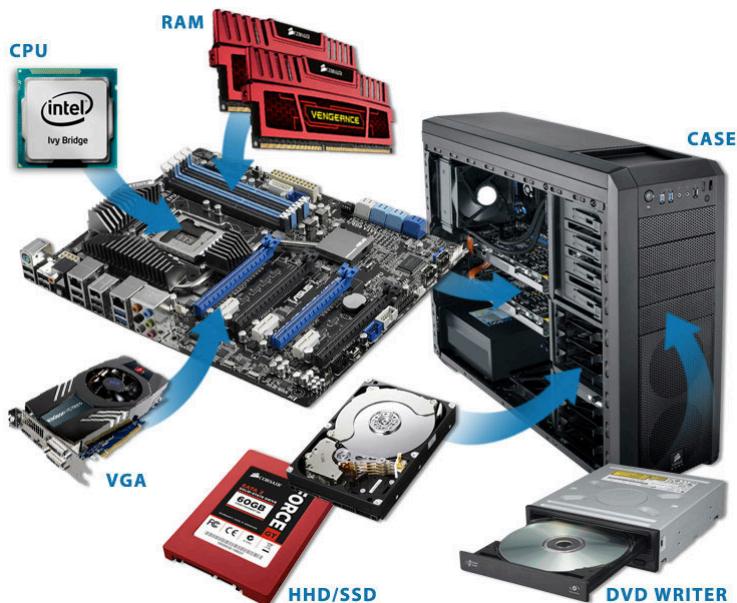
1. [Windows POSIX?](#)

2. [Linux POSIX?](#)

La configuration type d'un ordinateur

La configuration type d'un ordinateur comprend les éléments suivants :

- **Processeur (CPU)** : centralise et exécute les instructions (ex. Intel Core i5, AMD Ryzen 5).
- **Mémoire vive (RAM)** : stocke temporairement les données en cours de traitement (ex. 8 Go, 16 Go, DDR3, DDR4, Fréquences).
- **Disque de stockage** : pour stocker les données (ex. SSD 256 Go, HDD 1 To).
- **Carte graphique (GPU)** : gère l'affichage visuel et le traitement graphique (ex. Nvidia GTX 1650).
- **Carte mère** : connecte tous les composants entre eux.
- **Alimentation (PSU)** : fournit l'énergie électrique.
- **Système d'exploitation (OS)** : logiciel qui gère les ressources (ex. Windows, macOS, Linux).
- **Péphériques** : écran, clavier, souris, etc



Les différents types d'ordinateur

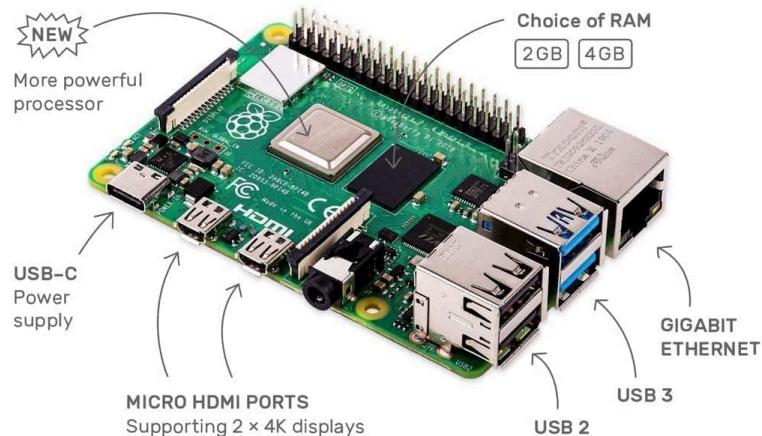
Il n'y a pas que les ordinateurs personnels

Quelques exemples

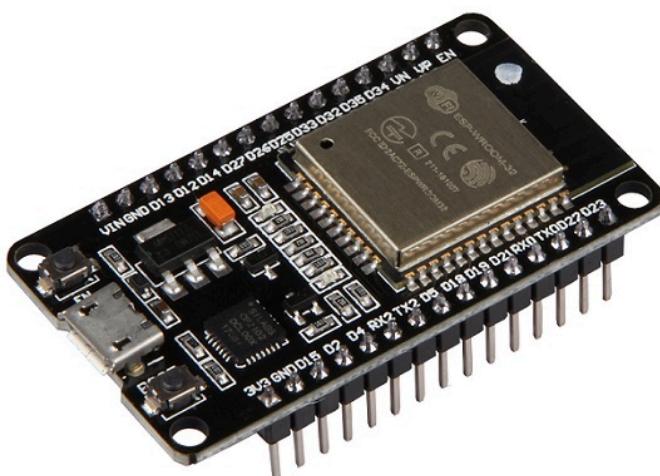
Serveur



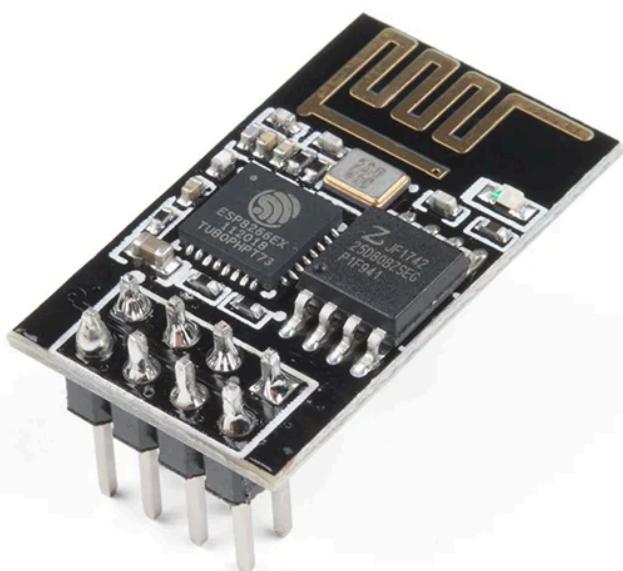
Raspberry



ESP64



ESP8266

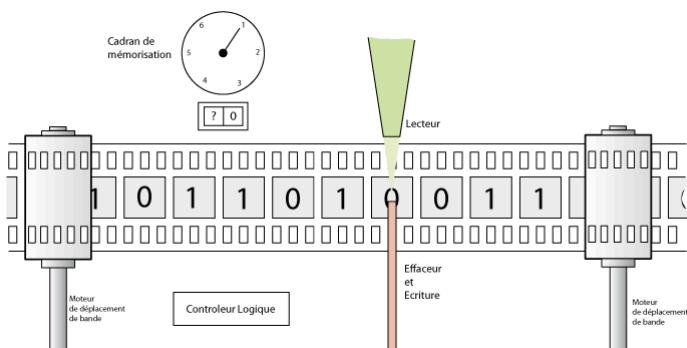




Les Principes de fonctionnement

Le traitement : le fonctionnement du processeur

C'est quoi une machine de turing?



Ruban et tête de lecture : La machine utilise un ruban infini qui sert de mémoire. La tête de lecture/écriture se déplace sur le ruban pour lire ou modifier les symboles inscrits (0, 1, ou un autre symbole).

États : La machine est toujours dans un état particulier. Il existe une série d'états prédéfinis, dont un état initial et un ou plusieurs états d'arrêt (ou finaux).

Instructions : Un ensemble d'instructions (programme) dicte ce que la machine doit faire en fonction de :

- Le symbole actuellement lu sur le ruban.
- L'état courant de la machine.

Opérations : En fonction des instructions, la machine peut :

- Écrire un symbole sur le ruban.
- Déplacer la tête vers la gauche ou la droite.
- Changer d'état.

Boucle : La machine continue à lire, écrire, se déplacer et changer d'état jusqu'à ce qu'elle atteigne un état d'arrêt, moment où elle termine l'exécution du programme.

Elle modélise de façon abstraite comment n'importe quel calcul algorithmique peut être effectué.

C'est avant tout un concept mathématique!

C'est quoi un processeur ?

Un processeur (CPU) est le composant central d'un ordinateur qui exécute les instructions des programmes en effectuant des calculs et en gérant les opérations logiques. Il agit comme le "cerveau" de l'ordinateur, coordonnant et contrôlant toutes les autres parties du système.

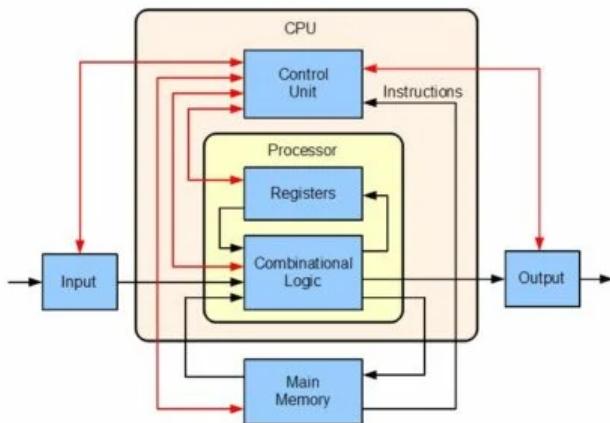
Il est lui même composé de différents éléments.

Les éléments d'un processeur

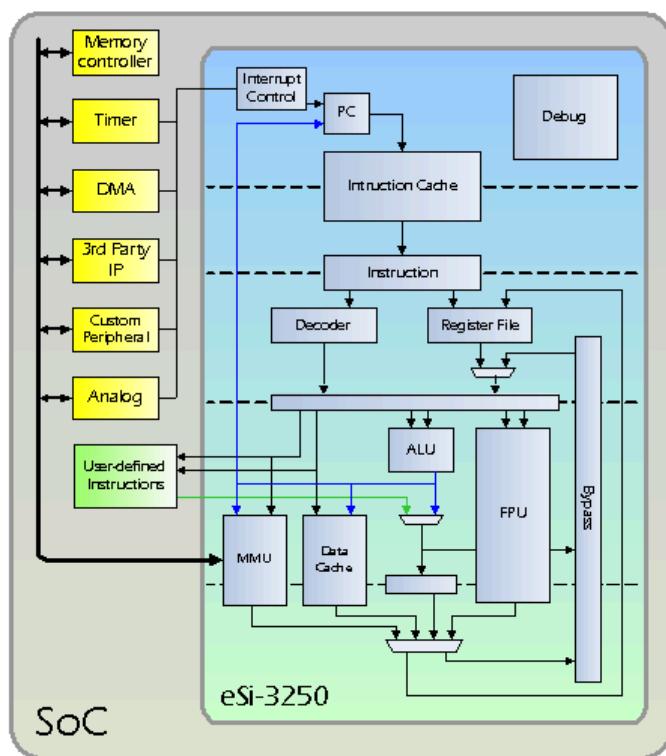
- **Unité de contrôle (Control Unit)** : dirige le flux de données et l'exécution des instructions en coordonnant les autres composants du processeur.
- **Unité arithmétique et logique (ALU)** : exécute les opérations mathématiques (addition, soustraction, etc.) et logiques (AND, OR, etc.).
- **Registres** : petites mémoires internes ultra-rapides qui stockent temporairement des données et des instructions pendant leur traitement.
- **Cache** : mémoire rapide intégrée au processeur, qui stocke des données fréquemment utilisées pour réduire le temps d'accès aux instructions et données.
- **Horloge (Clock)** : génère les impulsions de synchronisation pour coordonner l'exécution des instructions.
- **Unités de calcul spécialisées** : comme le **FPU (Floating Point Unit)** pour les calculs en virgule flottante, ou les **cores** (cœurs) multiples dans les processeurs multicœurs pour traiter plusieurs instructions en parallèle.

- **Bus internes** : canaux qui transportent les données, adresses et instructions entre les différents composants du processeur.

Une vue schématique



Une vue un peu plus complète (<https://fr.wikipedia.org/wiki/Processeur>)



Comment ça marche ?

Le fonctionnement d'un CPU (Central Processing Unit) repose sur plusieurs étapes clés appelées **cycle d'instruction** :

Fetch (Récupération) : Le CPU récupère une instruction depuis la mémoire (RAM), à une adresse spécifique, via un registre appelé *compteur de programme* (Program Counter, PC).

Decode (Décodage) : L'instruction récupérée est décodée par le décodeur d'instructions. Le CPU détermine de quel type d'instruction il s'agit (arithmétique, logique, contrôle, etc.) et quels opérandes (données) utiliser.

Execute (Exécution) : Le CPU exécute l'instruction décodée. Cela peut impliquer des calculs (addition, soustraction, etc.), des mouvements de données, ou des prises de décisions (branchements conditionnels).

Writeback (Écriture) : Les résultats de l'instruction sont écrits dans un registre, ou parfois en mémoire.

Ces étapes se répètent à grande vitesse (mesurée en GHz), permettant au CPU d'exécuter des milliards d'instructions par seconde.

C'est quoi des instructions? ou jeux d'instructions.

Un jeu d'instructions (ou *instruction set*) est l'ensemble des commandes ou instructions qu'un processeur peut comprendre et exécuter. Chaque instruction correspond à une opération spécifique (comme un calcul ou une gestion de données) que le processeur peut effectuer. Le jeu d'instructions inclut des commandes pour :

- Les opérations arithmétiques (addition, soustraction, etc.)
- Les opérations logiques (AND, OR, etc.)

- Le contrôle du flux (boucles, conditions)
- La gestion de la mémoire (chargement, stockage)

Chaque processeur à son jeux d'instruction. Il en existe plusieurs qui ont tous plusieurs versions.

Exemples de jeux d'instructions :

- AMD64 / x86-64
- IA-32 / x86
- ARM (Smartphone, console, Mac etc..)
- MIPS (systèmes embarqués)
- PowerPC (Ancien Apple)
- Itanium (server intel)

Exemple: <https://www.felixcloutier.com/x86/>

Core Instructions

Mnemonic	Summary
AAA	ASCII Adjust After Addition
AAD	ASCII Adjust AX Before Division
AAM	ASCII Adjust AX After Multiply
AAS	ASCII Adjust AL After Subtraction
ADC	Add With Carry
ADCX	Unsigned Integer Addition of Two Operands With Carry Flag
ADD	Add
ADDPD	Add Packed Double Precision Floating-Point Values
ADDPS	Add Packed Single Precision Floating-Point Values
ADDSD	Add Scalar Double Precision Floating-Point Values
ADDSS	Add Scalar Single Precision Floating-Point Values
ADDSUBPD	Packed Double Precision Floating-Point Add/Subtract
ADDSUBPS	Packed Single Precision Floating-Point Add/Subtract
ADOX	Unsigned Integer Addition of Two Operands With Overflow Flag
AESDEC	Perform One Round of an AES Decryption Flow
AESDEC128KL	Perform Ten Rounds of AES Decryption Flow With Key Locker Using 128-Bit Key
AESDEC256KL	Perform 14 Rounds of AES Decryption Flow With Key Locker Using 256-Bit Key
AESDECLAST	Perform Last Round of an AES Decryption Flow
AESDECWIDE128KL	Perform Ten Rounds of AES Decryption Flow With Key Locker on 8 Blocks Using 128-Bit Key
AESDECWIDE256KL	Perform 14 Rounds of AES Decryption Flow With Key Locker on 8 Blocks Using 256-Bit Key
AESENCL	Perform One Round of an AES Encryption Flow
AESENC128KL	Perform Ten Rounds of AES Encryption Flow With Key Locker Using 128-Bit Key
AESENC256KL	Perform 14 Rounds of AES Encryption Flow With Key Locker Using 256-Bit Key
AESENCLAST	Perform Last Round of an AES Encryption Flow
AESENCWIDE128KL	Perform Ten Rounds of AES Encryption Flow With Key Locker on 8 Blocks Using 128-Bit Key

ⓘ Info

Il est à noter qu'un jeu d'instructions ne peut pas être mis à jour. Il est inscrit physiquement dans le processeur.

Les autres composants d'un ordinateur

- **Carte Mère** : La carte principale d'un ordinateur qui connecte et communique avec tous les autres composants.
- **RAM (Mémoire Vive)** : Mémoire temporaire qui stocke les données en cours d'utilisation pour un accès rapide.
- **Disque Dur** : Dispositif de stockage de données à long terme.
- **Carte Graphique** : Composant spécialisé dans le traitement des graphismes et des images.

Le stockage : mémoires de masse, vive, morte, virtuelle, non volatile

Il existe plusieurs types de stockage:

Mémoire vive (RAM / Random Access Memory) :

- Type : volatile (perd les données à l'arrêt de l'ordinateur).
- Fonction : stockage temporaire des données en cours de traitement par le CPU.
- Caractéristiques : accès rapide, utilisée pour exécuter les programmes en cours.

Mémoire morte (ROM / Read-Only Memory) :

- Type : non volatile (conserve les données même après extinction).
- Fonction : contient les instructions essentielles au démarrage de l'ordinateur (comme le BIOS).
- Caractéristiques : lecture seule, ne peut pas être modifiée facilement.

Mémoire de masse :

- Type : non volatile.
- Fonction : stockage à long terme des fichiers, applications et systèmes d'exploitation.
- Types courants : disques durs (HDD), disques SSD.
- Caractéristiques : grande capacité de stockage, mais plus lente que la RAM.

Mémoire virtuelle :

- Type : temporaire (extension de la RAM).
- Fonction : utilisée lorsque la RAM est insuffisante, en transférant des données vers un espace de stockage sur disque dur.
- Caractéristiques : plus lente que la RAM, car elle repose sur le disque dur.

Mémoire non volatile (NVRAM) :

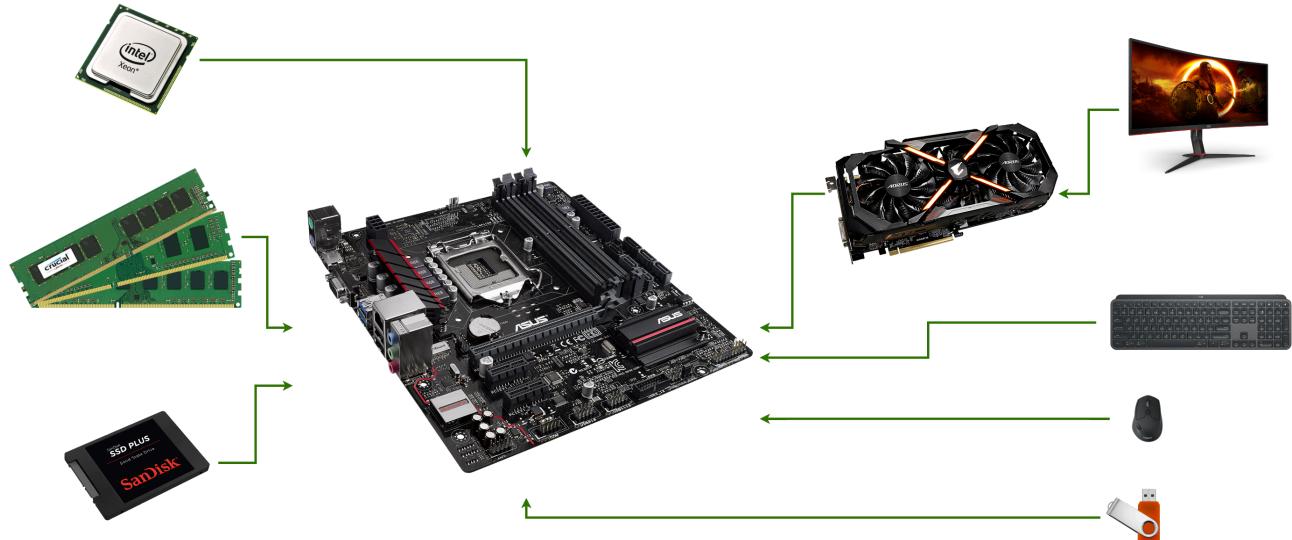
- Type : non volatile.
- Fonction : stocke des données importantes qui ne doivent pas être perdues (comme les paramètres du BIOS ou les données dans les SSD).
- Caractéristiques : conserve les données même sans alimentation électrique.

Ces mémoires varient en vitesse, capacité, et persistance, répondant à différents besoins dans l'architecture informatique.

Les périphériques

Il existe de nombreux périphériques possibles.

- Ecrans
- Clavier
- Souris
- Tablette graphique
- Clef USB
- Clef matériel
- Carte Bluetooth
- Carte Réseau
- Carte Son



Mais comment tout ça communique ?

La carte mère est responsable de manager la communication entre les différents éléments et périphériques. Pour cela elle utilise des bus de communication.

Un bus de communication est un ensemble de lignes (comprendre câblage) de transmission permettant l'échange de données entre les différents composants d'un ordinateur (processeur, mémoire, périphériques). Il transporte des signaux sous forme de données, d'adresses ou de commandes, facilitant ainsi la communication entre les parties du système.

- **Bus de Données** : Transfère les données entre les composants. Il s'agit d'un ensemble de fils électriques qui permettent le transfert d'informations.
- **Bus d'Adresse** : Indique où les données doivent être envoyées ou d'où elles doivent être récupérées dans la mémoire.
- **Bus de Commande** : Transmet les commandes entre le processeur et d'autres composants hardware.

Toutes ces communications ont besoin d'être synchronisées et doivent donc être aligné sur une horloge.

L'horloge dans un PC, appelée **horloge système** ou **clock**, est un composant qui génère des impulsions régulières (ou "ticks") pour synchroniser le fonctionnement de tous les composants du processeur et du système. Elle détermine la vitesse à laquelle le processeur exécute les instructions, mesurée en **hertz (Hz)**. Par exemple, un processeur cadencé à 3 GHz exécute 3 milliards de cycles par seconde.

Les composants peuvent s'enficher dans des slots ou connecteurs:

- **Slots et Connecteurs** : Permettent la connexion physique de composants supplémentaires comme la carte graphique, le disque dur, etc.

Pour coordonner tout ça:

- **Chipset** : Facilite la communication entre le processeur, la mémoire, les périphériques d'entrée/sortie, et d'autres composants.
- **Protocoles** : Des règles standardisées (comme SATA pour les disques durs, PCIe pour les cartes graphiques, USB pour les périphériques externes) assurent une communication cohérente
- **Le bios** : ...

Le binaire, les bits et les octets

Un ordinateur n'utilise que des chiffres et des nombres. Les signaux? des nombres. Les variables? des nombres. Des caractères alphanumérique? des nombres! Encore des nombres!

Pour tout ces nombres l'ordinateur ne connaît pas beaucoup de chiffre. Il n'en connaît que 2! Les 1 et les 0.

Il est probable que vous le sachiez déjà, c'est un fait connu aujourd'hui mais il faut en comprendre l'ampleur et notamment comprendre ce que cela implique.

Compter en binaire

Vous savez normalement compter en décimal. Soit utiliser 10 chiffres : 0.1.2.3.4.5.6.7.8.9.

Quand on arrive à 9, on ajoute un chiffre devant, d'ailleurs on peut considérer qu'il y a une infinité de 0 devant le chiffre. Exemple:
01.02.03.04.05.06.07.08.09.10.11.12.13...

Compter en binaire, c'est la même chose mais avec seulement 2 chiffres, les 0 et les 1. Exemple: 00, 01, 10, 11.

Comparons:

Décimal	Binaire
00	0000
01	0001
02	0010
03	0011
04	0100
05	0101
06	0110
07	0111
08	1000
09	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

En informatique

Dans le tableau précédent j'ai choisi de rajouter des 0 devant pour garder un affichage sur 4 chiffres. En informatique un ordinateur va plutôt en utiliser 8. Jamais moins. On appelle ça un **octet**. C'est devenu un standard en informatique qui permet d'assurer la compatibilité entre les différents systèmes.

Un **octet** est composé de 8 chiffres binaires, 8 **bits**.

Un octet peut prendre 256 valeurs, de 0 à 255.

Oui, le 0 est une valeur et vous n'avez pas fini d'y être confronté. Tout commence avec un 0 en informatique, pas à 1.

Décimal	Binanire
0	0000 0000
1	0000 0001
...	...
254	1111 1110
255	1111 1111

Tout est nombre dans un ordinateur, tout est binaire donc tout est **octet** ou multiple d'**octet**.

Info

Vous connaissez la table ASCII ? Chaque caractère est en fait un nombre binaire!

un exemple: <https://www.ascii-code.com/fr>

8bits, 16 bits, 32bits, 64 bits ...

Au fur et à mesure des années, les systèmes se sont complexifiés et ont donc eu besoin d'utiliser des plus grand nombre pour gérer tout ce qui se passe dans un ordinateur.

Il faut comprendre une chose primordiale. Si toutes les données sont gérées en octet et multiple d'octets, cela comprend aussi la manière dont on stock les données. Comment savoir où sont stockés les données ? physiquement! C'est simple, chaque emplacement à une adresse numéroté! Si votre système gère ces adresse en 8 bits, 16 bits ou 64 bits, alors il peut gérer plus d'adresse, donc plus de mémoire etc. Aussi bien dans la RAM, dans un disque dur ou encore partout où il y'en aura besoin.

Ce nombre de bits correspond à la taille des BUS dont on parle un peu plus haut.

Un BUS de 8 bits a une notion physique, les données circulent dans 8 câbles simultanément. Dans un système 64 bits, tous les BUS font 64 bits donc tous les BUS ont 64 "câbles" ou "pistes électroniques" pour faire circuler l'information plus rapidement.

Info

Cela à pris des années pour passer de systemes 8 bits aux 64 bits d'aujourd'hui.

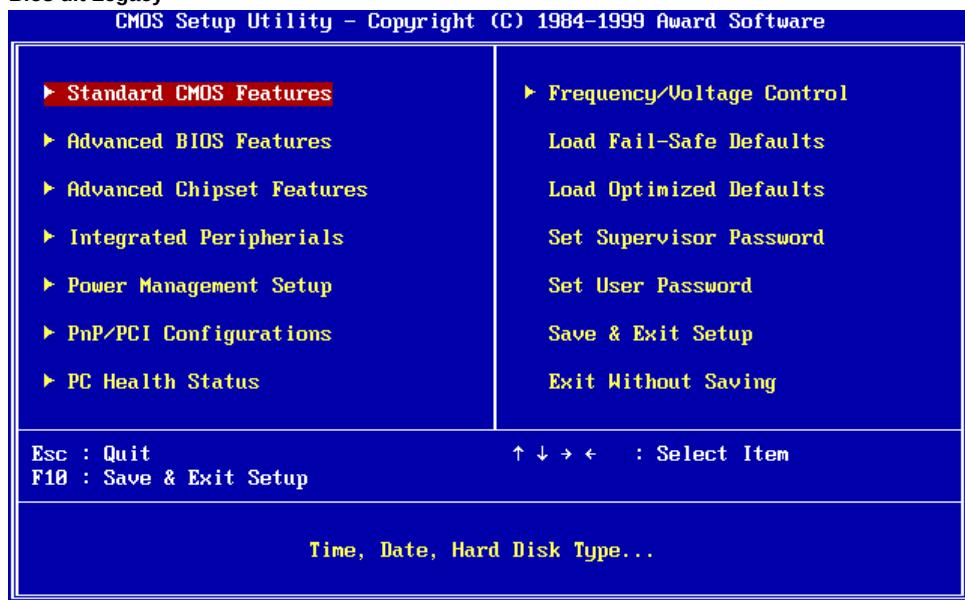
- **8 bits** : Années 1970.
- **16 bits** : Années 1980.
- **32 bits** : Années 1980-1990.
- **64 bits** : Années 2000 et après.

test

Le bios

Le **BIOS** (Basic Input/Output System) est un programme stocké dans la mémoire morte (ROM) d'un ordinateur, qui s'exécute au démarrage. Il initialise le matériel (processeur, mémoire, périphériques) et lance le système d'exploitation en chargeant son amorce (bootloader). Le BIOS fournit également une interface pour configurer certains paramètres matériels de la machine.

Bios dit Legacy



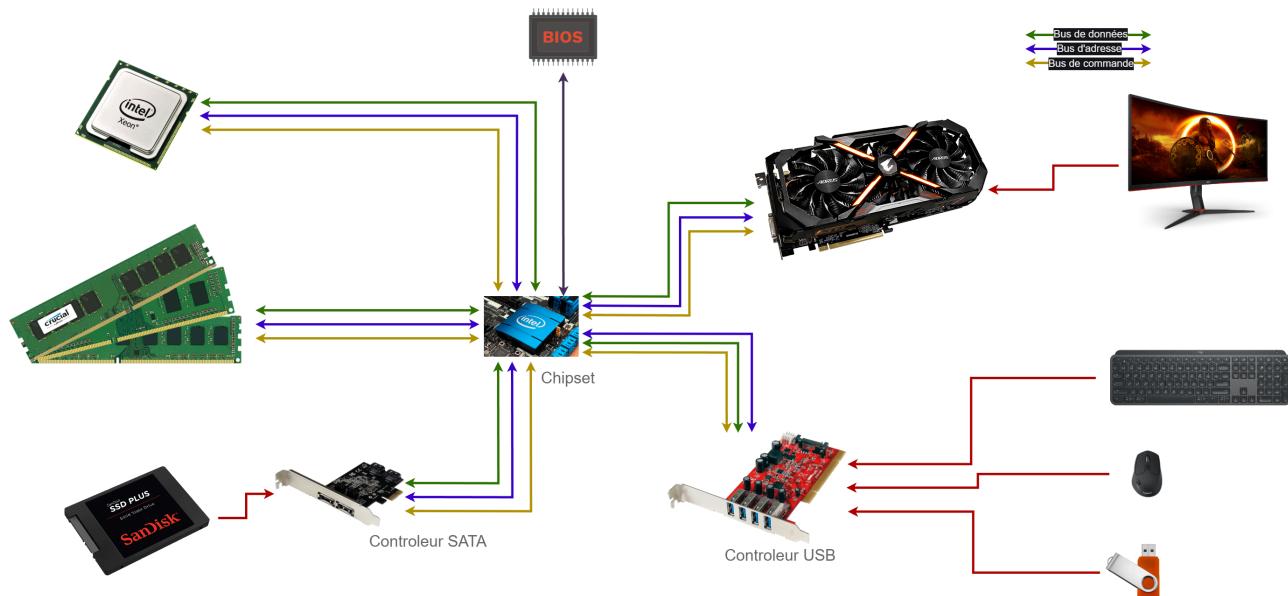
Bios dit UEFI



Exemple d'évolution de UEFI :

- Prise en charge de fonctionnalités réseaux
- Prise en charge du 64bits
- Prise en charge des disques supérieurs à 2,2To (jusqu'à 9,4 Zeta octets)
- Espace mémoire plus grand
- Prise en charge du secure boot

- Démarrage plus rapide



Secure Boot

Vérification des Signatures : Secure Boot vérifie les signatures numériques des logiciels, notamment le chargeur de démarrage (bootloader) et le système d'exploitation, pour s'assurer qu'ils n'ont pas été modifiés par un malware.

Blocage des Logiciels Non Autorisés : Si un logiciel n'a pas une signature valide, Secure Boot empêche son exécution, protégeant ainsi l'ordinateur contre des démaragements malveillants.

Configurable : Les utilisateurs peuvent gérer les paramètres de Secure Boot dans l'UEFI, y compris l'ajout de leurs propres clés de sécurité pour les logiciels personnalisés.

Les Drivers et Firmware

Pour pouvoir échanger via des protocoles commun, les périphériques et votre système d'exploitation , il est nécessaire d'ajouter une couche logiciel. Les drivers et les firmwares.

Drivers : Les drivers, ou pilotes, sont des programmes logiciels qui permettent au système d'exploitation d'un ordinateur de communiquer et de contrôler efficacement le matériel spécifique de l'ordinateur, comme les cartes graphiques, les imprimantes ou les périphériques USB.

Firmware : Le firmware est un type de logiciel intégré directement dans un composant matériel, comme la carte mère ou un disque dur. Il contient les instructions de base pour que le matériel puisse exécuter ses fonctions essentielles et communiquer avec d'autres parties de l'ordinateur.

Les disques

Pour stocker des données de manière durable ont parlé de Disque. Aussi bien mécanique avec les HDD ou électronique avec les SSD ou plus récemment les NVMe.

Les données sont stocké sous forme binaire, des bits (1 ou 0), ces données sont regroupés en divers tailles ou concepts:

Bits et octets

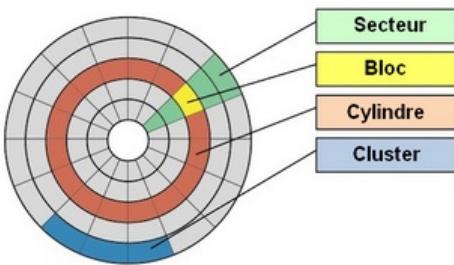
- **Bits** : Les données sont stockées sous forme de bits, les plus petites unités d'information (0 ou 1).
- **Octets** : 8 bits forment un octet, qui est l'unité de base pour stocker des caractères (ex. une lettre dans un texte).

Secteurs

- Les disques durs sont divisés en **secteurs**, chacun contenant une petite quantité de données, généralement **512 octets** ou **4096 octets** (4 Ko) dans les disques modernes.
- Les secteurs sont les plus petites unités de stockage physique sur un disque dur.

Clusters

- Un **cluster** est un regroupement de secteurs utilisé par le système de fichiers pour gérer les données. Les fichiers sont stockés dans un ou plusieurs clusters.
- La taille des clusters est définie lors de la création du système de fichiers et affecte l'efficacité du stockage.



Partitions

- Un disque dur peut être divisé en **partitions**, qui sont des sections logiques du disque. Chaque partition peut avoir son propre **système de fichiers**.
- Les partitions sont souvent utilisées pour séparer des systèmes d'exploitation ou pour organiser des données différemment.
- **Partition primaire** : utilisée pour installer un système d'exploitation.
- **Partition étendue** : peut contenir plusieurs sous-partitions logiques.

Système de fichiers

- Le **système de fichiers** organise et gère la façon dont les données sont stockées et récupérées sur une partition. Il permet au système d'exploitation de localiser les fichiers et de gérer l'espace disque.
- Différents systèmes de fichiers existent :
 - **FAT32** : ancien, compatible mais limité en taille de fichiers (max 4 Go).
 - **NTFS** : utilisé par Windows, supporte les grands fichiers et offre plus de sécurité et de gestion.
 - **EXT4** : utilisé sur Linux, performant et robuste.
 - **APFS** : système de fichiers d'Apple, optimisé pour les SSD.

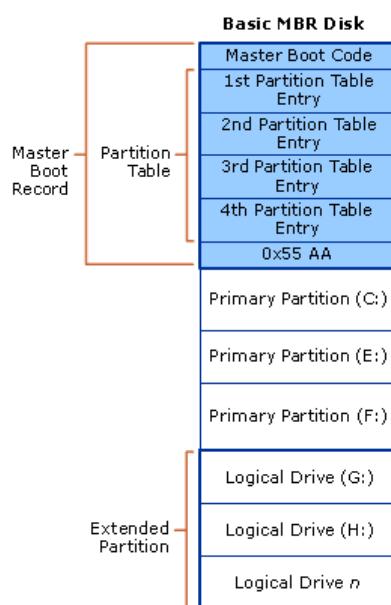
Stockage des types de données

- Les disques durs peuvent stocker différents types de données : **fichiers binaires** (programmes, images, etc.), **texte** (documents, code), **audio/vidéo**, etc.
- Le système de fichiers gère la répartition de ces données en clusters et secteurs pour une lecture/écriture efficace.

Quelques informations supplémentaire sur les partitions

MBR (Master Boot Record)

- **Ancien format** : Créé en 1983, il est utilisé sur les anciens systèmes et est compatible avec presque tous les systèmes d'exploitation.
- **Limite de taille** : Gère les disques jusqu'à 2 To maximum.
- **Nombre de partitions** : Permet la création de **4 partitions primaires** ou 3 partitions primaires et 1 partition étendue (qui peut contenir plusieurs partitions logiques).
- **Zone d'amorçage** : Le MBR contient un petit programme de démarrage (bootloader) qui charge le système d'exploitation. Ce programme est situé au début du disque.
- **Vulnérabilité** : Si le MBR est corrompu, le disque devient inutilisable car il contient les informations critiques de démarrage.

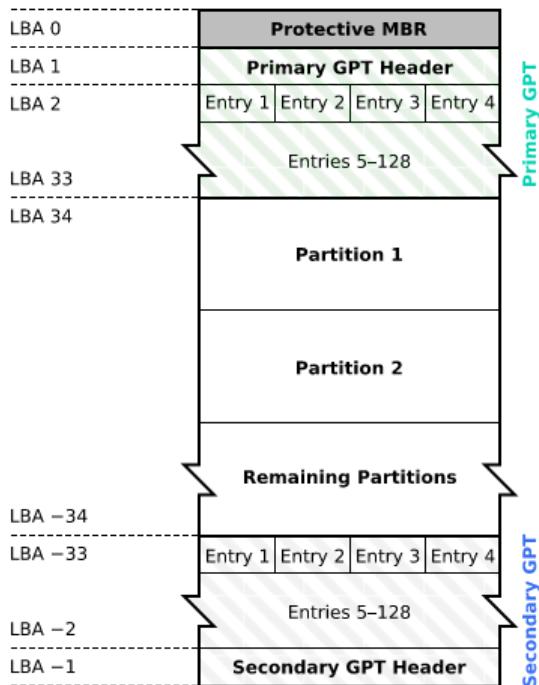


GPT (GUID Partition Table)

- **Format moderne** : Introduit avec l'UEFI (Unified Extensible Firmware Interface) pour remplacer le MBR, il est beaucoup plus puissant et flexible.

- **Limite de taille** : Gère les disques de plus de 2 To (jusqu'à 9.4 zettabytes).
- **Nombre de partitions** : Permet la création de jusqu'à **128 partitions** sans avoir besoin de partitions étendues.
- **Redondance** : GPT stocke plusieurs copies des informations de partition (au début et à la fin du disque), ce qui le rend plus résistant aux corruptions.
- **Identifiants uniques** : Chaque partition dans GPT a un **GUID (Globally Unique Identifier)**, garantissant une gestion unique des partitions.

GUID Partition Table Scheme



Différences clés :

- **Compatibilité** : MBR est plus ancien et largement pris en charge, mais GPT est recommandé pour les disques modernes et les systèmes UEFI.
- **Limite de partitions et de taille** : GPT est plus flexible avec un plus grand nombre de partitions et la gestion des disques de grande taille (>2 To).
- **Sécurité et fiabilité** : GPT offre une meilleure résistance aux corruptions grâce à sa redondance.

test

Le système d'exploitation (OS)

Operating Système ou Système d'Exploitation : Logiciel principal qui gère l'ensemble du matériel et des logiciels d'un ordinateur. Il permet la communication entre les programmes et le matériel et organise les fichiers, les ressources, et les tâches.

Exemple d'OS

- **Windows** : Développé par Microsoft, largement utilisé dans les environnements personnels et professionnels.
- **MacOS** : Système d'exploitation développé par Apple, spécifiquement pour les ordinateurs Mac.
- **Linux** : Un système d'exploitation open-source, populaire dans les serveurs et les systèmes embarqués, ainsi que chez les utilisateurs avancés.
- **Android** : Basé sur le noyau Linux, c'est le système d'exploitation mobile le plus utilisé, développé principalement par Google.
- **iOS** : Le système d'exploitation mobile développé par Apple, utilisé sur les iPhone et les iPad.

Composition d'un OS

Noyau (Kernel)

- **Fonction** : Cœur de l'OS, il gère les interactions entre le matériel et les logiciels.
- **Rôles** : Gestion des processus, de la mémoire, des périphériques et des fichiers. Il supervise l'exécution des programmes et l'allocation des ressources.

Gestion des processus

- **Fonction** : Coordonne l'exécution des programmes (processus) en allouant le temps processeur et gérant la priorité des tâches.

- **Sous-parties :**
 - **Ordonnanceur (Scheduler)** : Gère l'exécution des processus.
 - **Multitâche** : Permet l'exécution simultanée de plusieurs processus.

Gestion de la mémoire

- **Fonction** : Contrôle l'allocation de la mémoire vive (RAM) entre les applications et gère la mémoire virtuelle (utilisant le disque dur).
- **Sous-parties :**
 - **Mémoire virtuelle** : Étend la RAM en utilisant l'espace disque.
 - **Mémoire cache** : Stocke temporairement les données fréquemment utilisées pour améliorer la vitesse d'accès.

Différence entre Programme et Processus

- **Programme** : C'est un ensemble d'instructions stocké sur un disque ou un autre support. Il s'agit d'un fichier statique, souvent sous forme de code exécutable (ex. .exe), qui ne s'exécute pas encore. Un programme n'occupe pas de ressources mémoire ou processeur tant qu'il n'est pas lancé.
- **Processus** : C'est un programme en cours d'exécution. Une fois lancé, un programme devient un processus, qui est actif et utilise des ressources du système (mémoire, processeur). Il possède un **état**, des **variables**, et peut créer des sous-processus.

En résumé, un programme est un code statique, tandis qu'un processus est ce même code en exécution, utilisant des ressources du système.

Système de fichiers (File System)

On en a déjà parlé plus haut... mais on en remet une couche!

- **Fonction** : Organise et gère le stockage des données sur les disques.
- **Rôles** : Création, suppression, lecture et écriture de fichiers et de répertoires. Il fournit aussi une structure pour naviguer et accéder aux fichiers.
- Quelques File System connus:
 - **NTFS** (New Technology File System) : Utilisé principalement par les systèmes d'exploitation Windows pour le stockage sur disque dur.
 - **FAT32** (File Allocation Table 32) : Ancien système de fichiers utilisé par Windows, mais encore courant pour les lecteurs flash et autres supports de stockage amovibles.
 - **exFAT** (Extended File Allocation Table) : Conçu pour les supports amovibles comme les cartes SD et les clés USB, il est compatible avec les systèmes Windows et macOS.
 - **Ext4** (Fourth Extended Filesystem) : Système de fichiers utilisé principalement par les distributions Linux, connu pour sa robustesse et son efficacité.
 - **APFS** (Apple File System) : Utilisé par les systèmes d'exploitation modernes d'Apple, comme macOS, iOS, tvOS, et watchOS, optimisé pour les SSD et les fonctionnalités de sécurité et de cryptage.

Pilotes de périphériques (Device Drivers)

- **Fonction** : Logiciels spécifiques qui permettent à l'OS de communiquer avec les périphériques matériels (imprimantes, claviers, disques durs, cartes graphiques, etc.).
- **Rôles** : Gèrent les interactions entre le noyau et les périphériques externes.

Interface utilisateur

- **Fonction** : Permet à l'utilisateur d'interagir avec le système d'exploitation.
- **Sous-parties :**
 - **Interface en ligne de commande (CLI)** : Interaction via des commandes textuelles (ex. terminal, shell).
 - **Interface graphique (GUI)** : Interaction via des éléments visuels (fenêtres, icônes, menus).

Gestion des entrées/sorties (I/O Management)

- **Fonction** : Coordonne la communication entre les périphériques d'entrée (clavier, souris) et de sortie (écran, imprimante).
- **Rôles** : Gère les opérations de lecture et d'écriture sur les différents périphériques.

Services systèmes et utilitaires

- **Fonction** : Programmes ou services qui facilitent la gestion et la maintenance du système (gestion des utilisateurs, sécurité, logs, etc.).
- **Exemples** : Antivirus, gestionnaire de tâches, services réseau.

Les processus

Un processus est l'instance d'un programme en cours d'exécution.

Il a accès à des ressources (allouées)

Il a son propre espace d'adressage, géré par le système d'exploitation

Mécanismes de Communication inter processus:

- Pipes : Canaux unidirectionnels ou bidirectionnels pour passer des données entre processus.
- Files d'Attente de Messages : Permettent aux processus d'envoyer et de recevoir des messages de manière asynchrone.
- Mémoires Partagées : Un espace de mémoire accessible par plusieurs processus pour un accès rapide aux données.
- Sémaphores : Utilisés pour la synchronisation, permettant de contrôler l'accès aux ressources partagées.
- Signaux : Des notifications envoyées à un processus pour lui indiquer un événement particulier.

Fonctionnement :

- Système d'Exploitation : L'OS fournit les API et les mécanismes nécessaires pour mettre en place l'IPC.
- Synchronisation : Des mécanismes comme les verrous et les sémaphores assurent une communication cohérente sans conflit d'accès.
- Echange de Données : Les processus peuvent échanger des données soit par copie (messages), soit par accès direct (mémoire partagée).

Les API

API (Application Programming Interface) : Ensemble de règles et de spécifications que les logiciels peuvent suivre pour communiquer entre eux, permettant ainsi l'intégration et l'interaction de différents systèmes et applications.

Les API Web

- REST (Representational State Transfer) : Un style architectural pour les services web, utilisant des méthodes HTTP standard.
- SOAP (Simple Object Access Protocol) : Un protocole basé sur XML pour l'échange d'informations structurées dans l'implémentation de services web.
- GraphQL : Un langage de requête pour les API qui permet aux clients de demander exactement les données dont ils ont besoin.
- JSON-RPC et XML-RPC : Des protocoles permettant des appels de procédure distants, utilisant respectivement JSON et XML pour encoder les appels et les réponses.
- OData (Open Data Protocol) : Un standard basé sur des technologies web comme HTTP pour publier, modifier et interroger des données.

Les API Windows

- WinAPI (Windows API) : L'ensemble principal d'interfaces de programmation d'application Microsoft Windows, utilisé pour les interactions au niveau du système d'exploitation.
- COM (Component Object Model) : Un modèle de composant logiciel Microsoft utilisé pour permettre l'interaction et la réutilisation des objets logiciels au sein de Windows.
- OLE (Object Linking and Embedding) : Utilisé pour l'intégration et la communication entre différentes applications sous Windows, permettant par exemple d'incorporer un tableau Excel dans un document Word.
- ActiveX : Un ensemble de technologies basées sur COM pour la création de composants interactifs pour les applications Internet et Windows.
- DirectX : Une collection d'API pour gérer les tâches liées aux jeux et à la vidéo sur la plateforme Windows, notamment pour la programmation de graphiques et de sons.

Ordonnancement des processus

L'ordonnancement des processus dans un système d'exploitation est le mécanisme par lequel l'OS décide quel processus doit être exécuté par le processeur à quel moment et pour combien de temps. Ce processus est crucial pour la gestion efficace des ressources du CPU, en particulier dans les systèmes multitâches

File d'Attente des Processus : Les processus prêts à être exécutés sont placés dans une file d'attente.

Choix du Processus : L'ordonnanceur du système d'exploitation sélectionne un processus de la file d'attente selon une certaine politique d'ordonnancement.

Politiques d'Ordonnancement :

- First-Come, First-Served (FCFS) : Les processus sont exécutés dans l'ordre de leur arrivée.
- Plus Court d'Abord (Shortest Job First - SJF) : Priorité aux processus avec le temps d'exécution le plus court.
- Round Robin (RR) : Chaque processus reçoit une petite unité de temps (quantum); si un processus n'a pas terminé à la fin de son quantum, il est remplacé à la fin de la file d'attente.
- Priorité : Les processus avec une priorité plus élevée sont exécutés en premier.

Changement de Contexte : Lorsqu'un processus est interrompu (par exemple, à la fin de son quantum en Round Robin), l'OS sauvegarde son état (contexte) avant de passer au processus suivant.

Gestion de la Concurrence : L'ordonnanceur doit également gérer les dépendances entre processus, comme l'attente de ressources ou la communication inter-processus.

POSIX

Portable Operating System Interface (oui y'a pas de X, juste une ref a l'héritage d'UNIX).

Pas un OS mais une famille de normes techniques depuis 1988

<https://fr.wikipedia.org/wiki/POSIX>

Exemples de règle :

- Égalité des Processus : Sous POSIX, il n'y a généralement pas de distinction entre un "processus maître" et d'autres processus en termes d'ordonnancement. Tous les processus sont soumis aux mêmes règles d'ordonnancement.
- Priorité et Ordonnancement : L'ordonnancement est basé sur la priorité et d'autres métriques, plutôt que sur une hiérarchie de processus.
- Fork et Exécution : POSIX utilise souvent le modèle "fork-exec" pour la création de nouveaux processus, où un processus parent (fork) crée un processus enfant qui peut exécuter un programme différent. Le processus parent n'est pas nécessairement un "maître" sur l'enfant en termes d'ordonnancement.
- Signaux et Communication : Bien qu'un processus puisse envoyer des signaux ou communiquer avec d'autres processus, cela ne lui confère pas un statut de "maître".

Windows POSIX?

Architecture Différente : Windows est construit sur une architecture distincte de celle des systèmes UNIX, sur lesquels POSIX est basé.

Gestion des Processus : Windows utilise un modèle de gestion des processus et des threads différent de POSIX.

API Windows vs POSIX : L'interface de programmation de Windows (WinAPI) diffère significativement de l'API POSIX.

Outils et Commandes Spécifiques : Les outils en ligne de commande et les utilitaires sous Windows ne correspondent pas à ceux de UNIX/POSIX.

Compatibilité Limitée : Bien que Windows ait tenté d'intégrer une compatibilité POSIX, celle-ci reste partielle et peu utilisée.

Linux POSIX?

De moins en moins avec l'arrivée de SystemD

Objectif : Systemd vise à fournir une gestion efficace et moderne des processus et services au démarrage, ce qui inclut le lancement et la supervision des services, la gestion des dépendances entre services, et d'autres fonctionnalités avancées.

Fonctionnalités Spécifiques à Linux : Systemd utilise de nombreuses fonctionnalités spécifiques au noyau Linux qui ne sont pas couvertes par la norme POSIX, comme les cgroups pour la gestion des ressources.

Compatibilité avec POSIX : Bien que systemd fonctionne sur des systèmes Linux qui peuvent être en grande partie compatibles avec POSIX, systemd lui-même n'est pas conçu pour respecter ou implémenter directement les standards POSIX.

Approche et Design : Systemd adopte une approche et un design différents pour la gestion des processus et des services, se concentrant sur les besoins des systèmes Linux modernes plutôt que sur la conformité aux normes POSIX.