

## Matériels, systèmes et logiciels

---

### Introduction :

Les appareils électroniques et informatiques modernes sont très sophistiqués : les smartphones actuels embarquent une puissance de calcul bien supérieure à celle des volumineux ordinateurs d'anciennes générations. Matériels, systèmes et logiciels ont beaucoup évolué depuis les débuts de l'informatique, et les capacités de communication des appareils actuels sont sans commune mesure avec celles des générations précédentes.

En dépit de ces modernisations et des changements de forme et de taille, le principe général de fonctionnement de tous ces appareils est resté assez identique, orchestré par un système d'exploitation qui pilote les ressources matérielles et logicielles mises à disposition de l'utilisateur·rice. Nous présenterons dans une première partie les principales architectures matérielles, puis nous nous intéresserons dans une seconde partie à la gestion des ressources par le système d'exploitation.

## 1 | Architectures matérielles

Les appareils informatiques ont grandement évolué depuis leur invention. Si leur puissance continue de s'accroître à un rythme élevé, la forme de ces appareils a également évolué. Les ordinateurs de bureau ont été déclinés dans des versions transportables, puis portables, à partir des années 1980 et 1990.

À la suite du développement des smartphones, à la fin des années 2000, le marché des tablettes a connu un essor notable dans les années 2010. Smartphones et tablettes constituent de véritables systèmes informatiques.



De nombreux autres appareils plus spécialisés, comme les consoles de jeux ou les récepteurs GPS, sont eux aussi des appareils informatiques.

En dépit de leur diversité de taille, de forme et d'usage, tous ces appareils modernes sont dérivés d'une architecture imaginée à la fin de la Seconde Guerre mondiale.

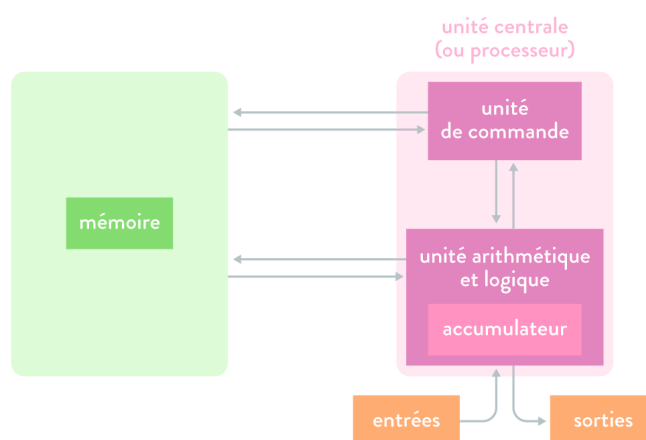
#### a. Architecture de von Neumann

Le mathématicien et physicien américano-hongrois John von Neumann a conceptualisé en 1945 un ordinateur stockant un programme en mémoire, inspiré des [machines de Turing](#).

Présenté comme un « système de calcul numérique automatique à très haute vitesse », l'ordinateur de von Neumann est composé de cinq éléments :

- une unité de traitement ;
- une unité de contrôle ;
- une mémoire de travail ;
- des mécanismes d'entrées-sorties ;
- une mémoire de stockage externe.

## Modèle de von Neumann

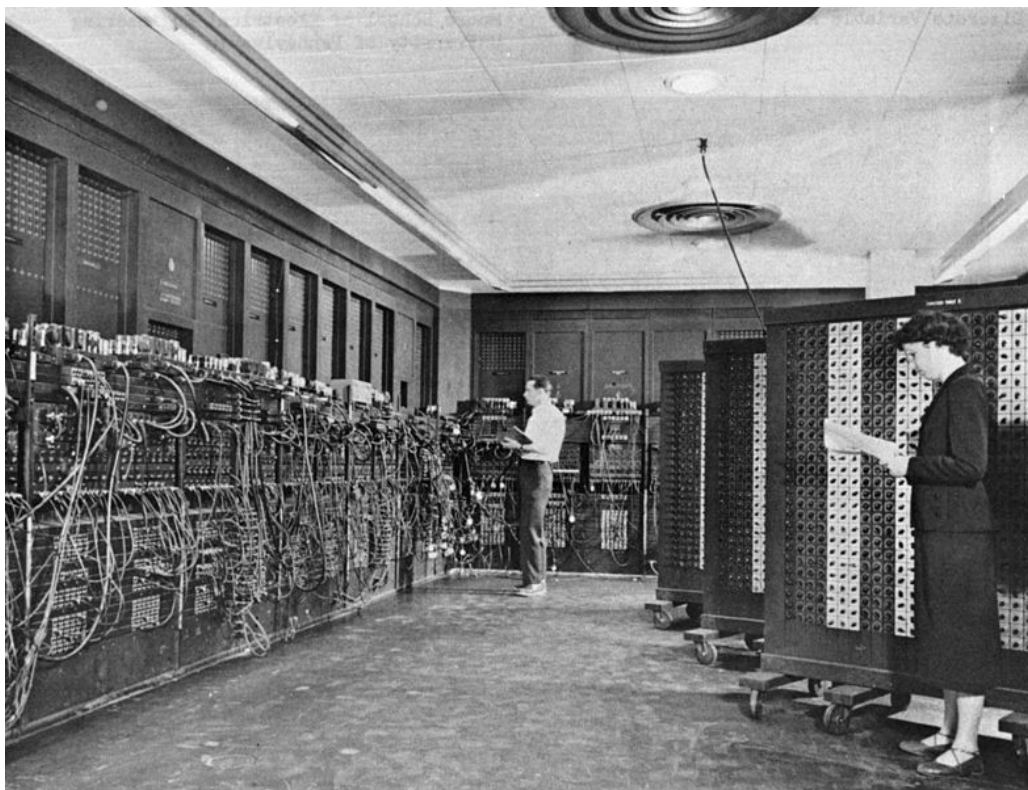


© SCHOOLMOUV

- 1 L'**unité de traitement** comprend une unité arithmétique et logique et des registres permettant d'effectuer des calculs.
- 2 L'**unité de contrôle** est la partie centrale qui pilote les suites d'instructions. Elle s'appelle également **l'unité de commande**.
- 3 La **mémoire** stocke les programmes et les données de travail.
- 4 Les **mécanismes d'entrées-sorties** constituent les interfaces qui permettent de communiquer avec l'ordinateur.
- 5 La **mémoire de stockage externe**, moins rapide que la mémoire de travail, sert à conserver durablement les programmes et les données.



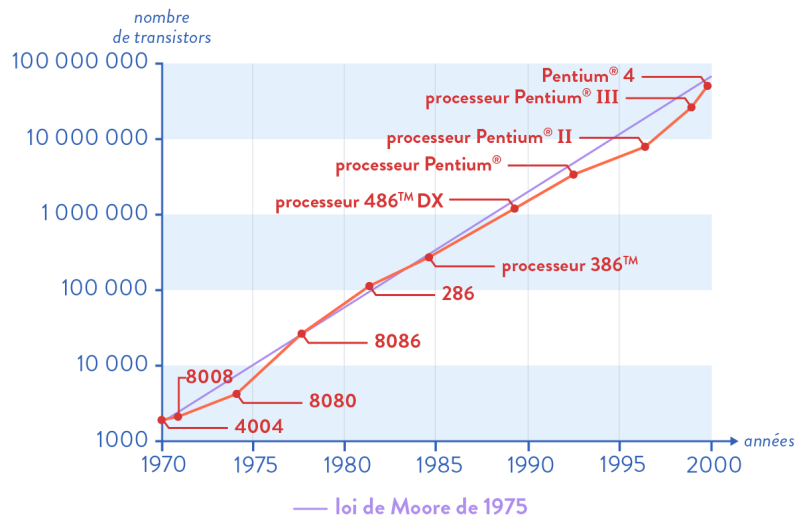
À l'époque où von Neumann propose cette architecture, les ordinateurs fonctionnent avec des interrupteurs, des relais et des tubes à vide. Les mémoires de stockage externe sur lesquelles les données sont enregistrées sont des cartes ou des rubans perforés.



L'apparition des **transistors** dans les années 1950, puis des premières **puces électroniques** ou **circuits intégrés** dans les années 1960, va apporter la puissance et la fiabilité qui faisaient défaut aux générations précédentes. Ces évolutions permettront en outre une miniaturisation toujours croissante des appareils.

#### **b.** Loi de Moore

En 1965, l'ingénieur américain Gordon Moore, co-fondateur de l'entreprise Fairchild Semiconductor et PDG de la société Intel à l'époque, constate que les progrès de la **miniaturisation** ont permis de doubler chaque année le nombre de transistors sur une surface donnée. Il prédit que ce doublement du nombre de transistors se poursuivra tous les deux ans, au moins sur une décennie.



© SCHOOLMOUV



Cette tendance exponentielle se poursuivra bien au-delà, et son affirmation empirique est appelée **loi de Moore**.

L'ingénieur expliquera toutefois, en 2005, que cette tendance ne pourra se produire indéfiniment. Notamment parce que les transistors ne pourront pas s'affranchir des limites de la miniaturisation au niveau atomique.

Les premiers ordinateurs occupaient des pièces entières. Les mini-ordinateurs des années 1960 à 1980 occupaient le volume d'une ou plusieurs étagères. Les micro-ordinateurs des années 1980 tenaient sur un bureau, et les premiers modèles transportables voyaient le jour.



Intéressons-nous maintenant à l'architecture d'un micro-ordinateur moderne.

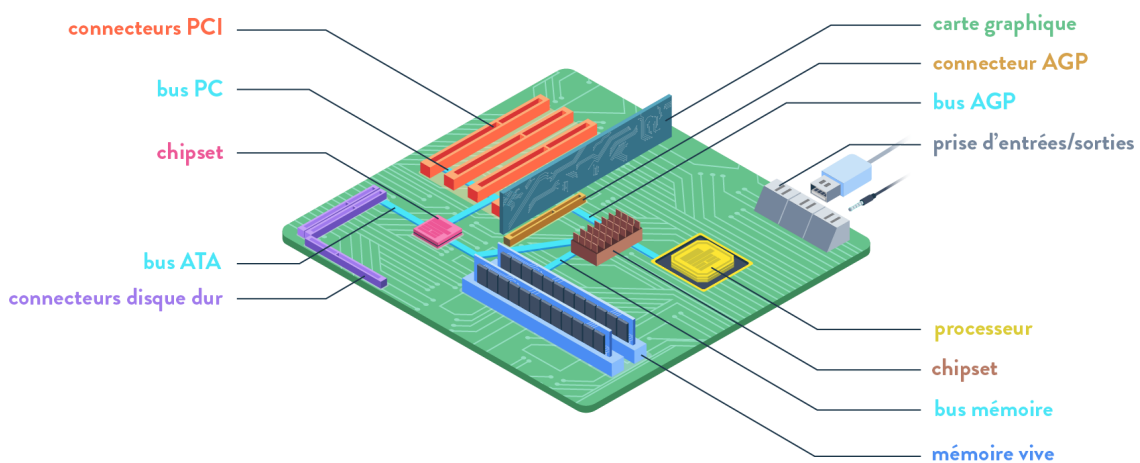
### c. Architecture d'un micro-ordinateur

Un **micro-ordinateur** est construit sur la base d'un grand circuit imprimé appelé **carte mère**. Ce circuit est destiné à recevoir et à connecter entre eux l'ensemble des composants nécessaires au fonctionnement de l'ordinateur.

La carte-mère d'un micro-ordinateur moderne comporte les éléments suivants :

- le micro-processeur ;
- la mémoire vive, généralement sous forme de barrettes modulaires ;
- une mémoire non volatile contenant les instructions d'amorçage de l'ordinateur ;

- différents connecteurs (alimentation électrique, mémoire de masse, entrées-sorties, cartes d'extension) ;
- des bus de données qui véhiculent les données entre les composants du micro-ordinateur.



© SCHOOLMOUV



La **connectique d'extension** est destinée à permettre une évolution modulaire de l'appareil, par l'ajout de composants individuels sous forme de cartes enfichables sur la carte électronique mère.

Les premiers micro-ordinateurs intégraient très peu de fonctionnalités en standard : il fallait ajouter une carte son pour disposer de fonctionnalités audio, une carte réseau pour disposer d'une connectivité (historiquement filaire), ou encore une carte contrôleur dédiée pour pouvoir disposer de ports USB sur la machine, chaque carte occupant un emplacement d'extension.

→ L'essor des **usages multimédia** et des **réseaux** a conduit à l'intégration de bon nombre de ces fonctionnalités en standard sur les cartes mères modernes.

→ La **miniaturisation des composants** a également permis la conception de systèmes complets sur puce unique, notamment dans le domaine industriel.

Cette architecture est également employée dans des appareils omniprésents de la vie quotidienne, personnelle ou professionnelle : les ordiphones, également appelés smartphones ou téléphones intelligents.

#### d. Architecture d'un smartphone

L'architecture d'un smartphone est construite autour d'un **système sur puce**.



#### Définition

##### Systeme sur puce :

Un système sur puce est un circuit intégré réunissant les composants habituellement séparés d'un ordinateur ou d'un appareil électronique.



#### Astuce

Ces systèmes sur puce sont couramment désignés par le sigle **SoC** (pour *System on Chip* en anglais, soit « système sur puce »).

Un SoC comporte généralement :

- au moins un processeur ;
- éventuellement des co-processeurs ;
- de la mémoire, qui peut être de différents types ;
- des fonctions de traitement du signal ;
- des interfaces d'entrées-sorties.

D'autres composants spécialisés peuvent être intégrés en fonction des besoins et du type d'appareil qui embarque le système sur puce.



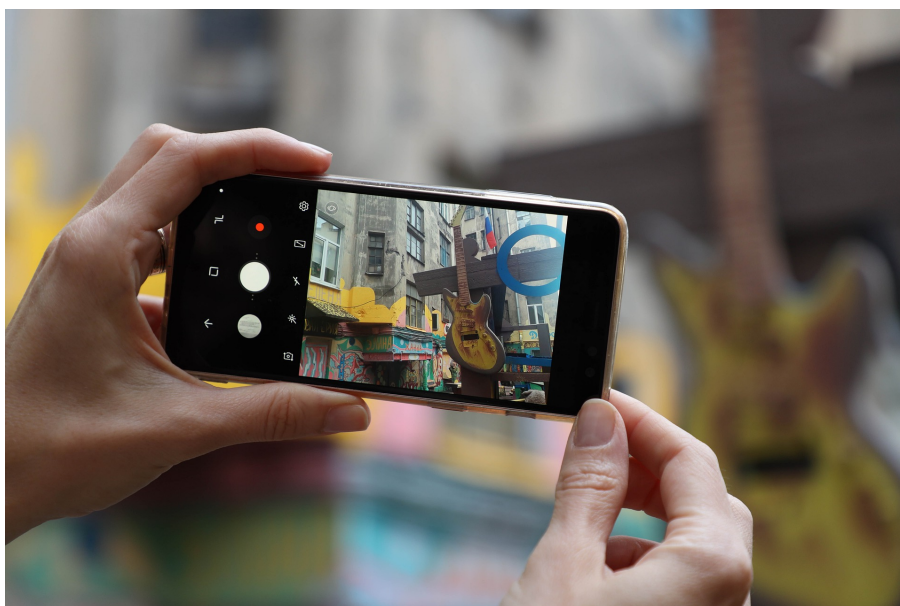
L'intégration des composants sur une même puce présente **plusieurs avantages** par rapport à l'architecture dissociée du micro-ordinateur :

- meilleures performances ;
- plus grande compacité ;
- moindre consommation électrique ;
- meilleure dissipation thermique ;
- moins de matières premières consommées ;
- coût de fabrication réduits.

→ Cette meilleure intégration se fait au prix d'un coût de développement important et d'une flexibilité amoindrie.

Les systèmes sur puce sont utilisés dans un grand nombre d'appareils et systèmes électroniques : véhicules, récepteurs GPS, Internet des objets, appareils industriels, mais aussi smartphones.

Les téléphones intelligents ont en effet besoin d'intégrer de la puissance et de nombreuses fonctionnalités dans un espace très réduit en taille et peu ventilé, tout en consommant le moins d'énergie électrique possible pour pouvoir fonctionner longtemps sur batterie.



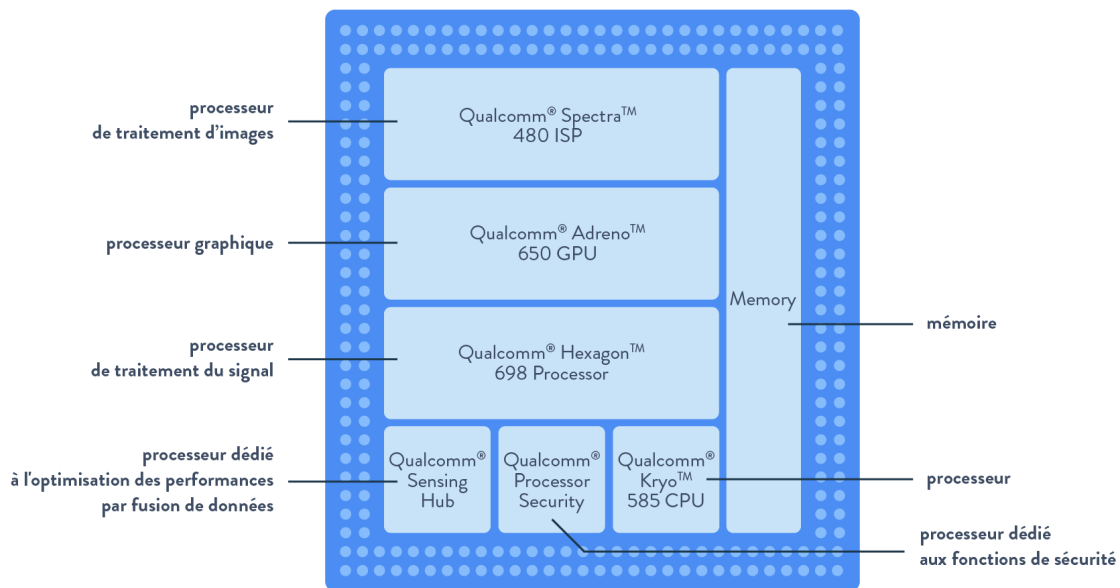
L'entreprise américaine Qualcomm Technologies et l'entreprise coréenne Samsung Electronics sont d'importants producteurs de systèmes sur puce pour smartphone.

Leurs systèmes sur puce intègrent de nombreux composants : un ou plusieurs micro-processeurs, un processeur graphique, des mémoires locales, des processeurs de traitement du signal et des interfaces de communication filaires et radio.



Voici les principaux composants du SoC Snapdragon 865 de Qualcomm Technologies :

- processeur **64 bits** Kryo 585 jusqu'à **2,84 GHz** ;
- jusqu'à **16 Go** de mémoire vive ;
- processeur graphique ;
- processeur de traitement du signal avec moteur d'intelligence artificielle ;
- processeur de traitement d'image avec supports des principaux codecs vidéo ;
- processeur dédié aux fonctions de sécurité ;
- modem radiofréquences LTE 5G ;
- connectivité Wi-Fi **2,4 GHz** et **5 GHz** ;
- connectivité Bluetooth ;
- système de charge intelligente de la batterie ;
- système de positionnement par satellite (GPS, GLONASS, Beidou, Galileo) ;
- connectivité USB 3.



© SCHOOLMOUV

Les systèmes sur puce ne sont toutefois pas réservés aux smartphones ou aux appareils électroniques embarqués ou industriels.

→ Certains ordinateurs sont construits autour d'un système sur puce.



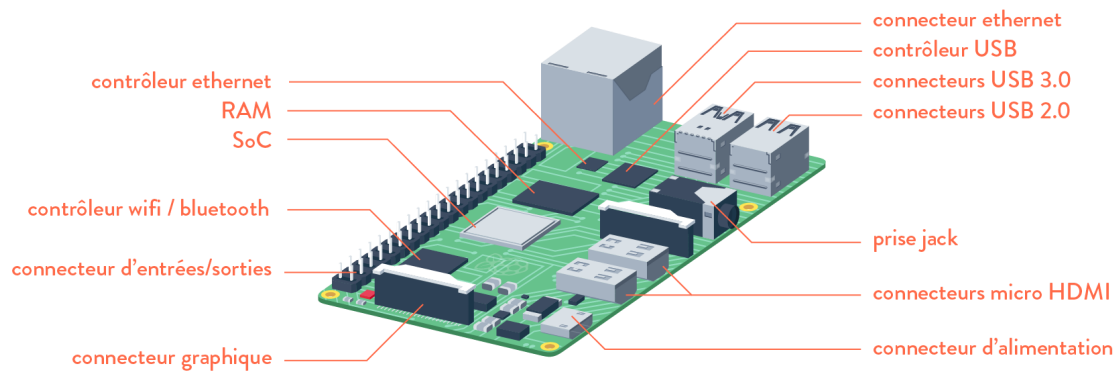
Exemple

C'est notamment le cas du Raspberry Pi, véritable ordinateur à petit prix, très utilisé dans un contexte pédagogique ou expérimental. Construit autour d'un SoC, le Raspberry Pi est un nano-ordinateur mono-carte vendu dépourvu de périphériques d'entrées sorties (comme peut l'être un serveur ou une unité centrale d'ordinateur de bureau) mais équipé en standard d'une connectivité USB et réseau, filaire et sans fil.



Astuce

Le Raspberry Pi est un nano-ordinateur qui a la taille d'une carte de crédit.



© SCHOOLMOUV

Nous avons présenté les architectures matérielles sur lesquelles reposent les ordinateurs et les smartphones. Nous allons maintenant nous intéresser aux systèmes d'exploitation, qui font le lien entre les ressources matérielles d'un dispositif électronique et les logiciels ou applications employés par les utilisateurs.

## 2 | Systèmes d'exploitation

Les ressources et fonctionnalités des ordinateurs et smartphones sont mises à disposition et gérées par un système logiciel appelé **système d'exploitation**. Comme son nom l'indique, il est destiné à permettre l'exploitation de l'appareil.

### a. Principaux systèmes d'exploitation

- Sur les **ordinateurs personnels**, le principal système d'exploitation est Microsoft Windows. Il équipe environ huit ordinateurs sur dix. Les autres ordinateurs personnels sont majoritairement équipés du MacOS d'Apple ou de distributions Linux.
- Les **serveurs web** sont majoritairement équipés de systèmes d'exploitation de la famille UNIX, notamment des distributions Linux.

- Sur les **smartphones**, le principal système d'exploitation est Google Android. Il équipe environ sept smartphones sur dix. Les autres smartphones sont majoritairement équipés d'Apple iOS.

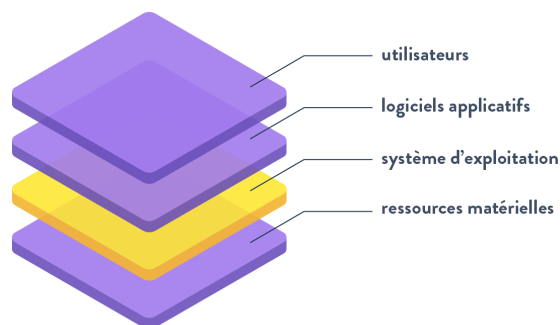
Comme les logiciels applicatifs, les systèmes d'exploitation peuvent être gratuits ou payants, libres ou propriétaires.

## **b.** Fonctions du système d'exploitation

Les systèmes d'exploitation remplissent plusieurs fonctions complémentaires.

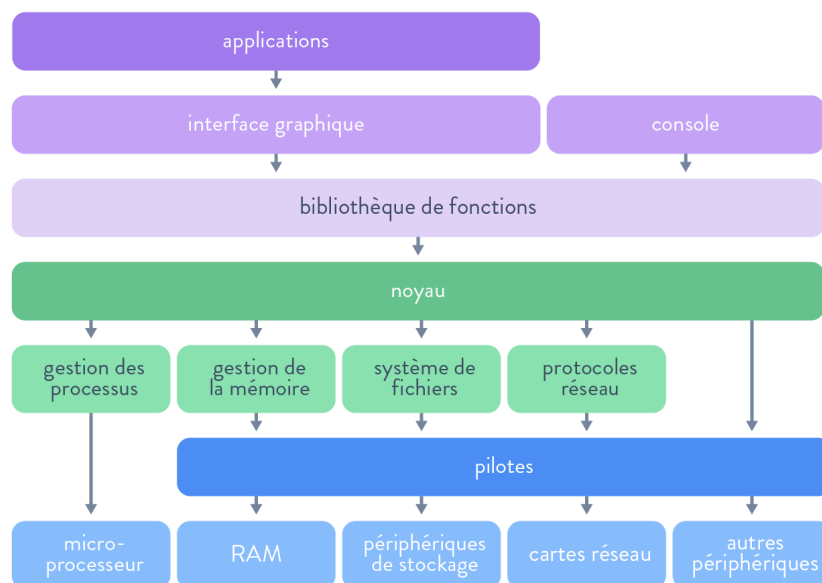
Un système d'exploitation :

- coordonne l'usage et les accès aux différentes ressources de l'appareil, notamment à la mémoire, au processeur, ainsi qu'aux périphériques ;
- assure le bon fonctionnement général du système, notamment la gestion des fichiers et la sécurité ;
- propose un socle de services communs aux logiciels et applications.



© SCHOOLMOUV

Le système d'exploitation constitue une **couche d'abstraction** par rapport aux ressources matérielles de l'appareil.



© SCHOOLMOUV

Différents **composants** assurent le bon fonctionnement du système d'exploitation.

### 1 Le **noyau** (*kernel* en anglais)

Il constitue le cœur du système d'exploitation. Il gère et coordonne le fonctionnement de l'ensemble des processus et des programmes, en organisant le nécessaire partage des ressources et des périphériques du système informatique.

### 2 Le **système de fichiers** (*file system* en anglais)

Il contrôle et organise le stockage en mémoire de masse et les accès aux fichiers. Des mécanismes de contrôle définissent les différents droits d'accès (lecture, écriture, exécution) d'un utilisateur individuel ou d'un groupe d'utilisateurs.

### 3 La gestion des **entrées-sorties**

Le système d'exploitation sert d'intermédiaire entre les logiciels et les équipements matériels. Il propose un accès logiciel simplifié aux périphériques par le biais de pilotes (*drivers* en anglais). Un pilote indique au système d'exploitation les spécificités de communication et de contrôle d'un périphérique donné.

## 4 L'interface utilisateur (*shell* en anglais)

Les interactions entre l'utilisateur et le système s'effectue par le biais d'une interface donnant accès à un certain nombre d'informations et de fonctionnalités du système. Certaines interfaces se présentent sous la forme de lignes de commande, d'autres sous forme d'une interface utilisateur graphique.

Les systèmes d'exploitation modernes intègrent également des fonctionnalités étendues au niveau réseau et sécurité.

### c. Gestion des processus



#### Définition

##### Processus :

Un processus (*process* en anglais) est un programme en cours d'exécution. Il se définit par plusieurs caractéristiques comme l'utilisation d'un espace mémoire pour stocker les données et l'exécution d'un ensemble d'instructions.



#### Attention

Il ne faut pas confondre un programme et ses processus : à un seul programme peuvent correspondre un à plusieurs processus, selon qu'il est exécuté par un ou plusieurs utilisateur·rice·s simultanément.

Les systèmes d'exploitation modernes sont à la fois :

- multi-tâches (ils permettent une exécution perçue comme simultanée de plusieurs actions) ;
- multi-utilisateurs (ils peuvent gérer plusieurs utilisateurs différents, disposant chacun de droits distincts) ;
- multi-processeurs (ils peuvent tirer profit de la présence éventuelle de plusieurs micro-processeurs).

Les ressources du système informatique, telles que le temps du processeur et l'espace de la mémoire, sont donc partagées entre différents processus pouvant appartenir à différents utilisateurs ou au système. **Ces partages nécessitent d'être coordonnés par le système**, pour des raisons d'efficacité, et pour éviter d'éventuels conflits d'accès ou d'utilisation de ressources.

Le système d'exploitation lance un certain nombre de processus nécessaires à son propre fonctionnement. Il permet également aux utilisateurs de demander le lancement de programmes, depuis la ligne de commande d'un terminal ou par l'activation d'une icône sur une interface graphique.

Le système d'exploitation attribue un identifiant unique à chaque processus. Il est possible de visualiser les processus en cours d'exécution et leurs identifiants avec les deux commandes UNIX suivantes :

- la commande **ps** (pour *process status*, soit « statut des processus »), qui affiche une liste statique des processus au moment de la requête ;
- la commande **top** qui présente une liste mise à jour dynamiquement des principaux processus et de leur état.

L'identifiant apparaît sous la colonne PID (pour *Process ID*, soit « identifiant du processus »). Les éventuels processus enfants référencent le processus parent dans la colonne PPID (pour *Parent Process ID*).

L'utilisateur à l'origine du processus est référencé par son identifiant UID (*User ID*).

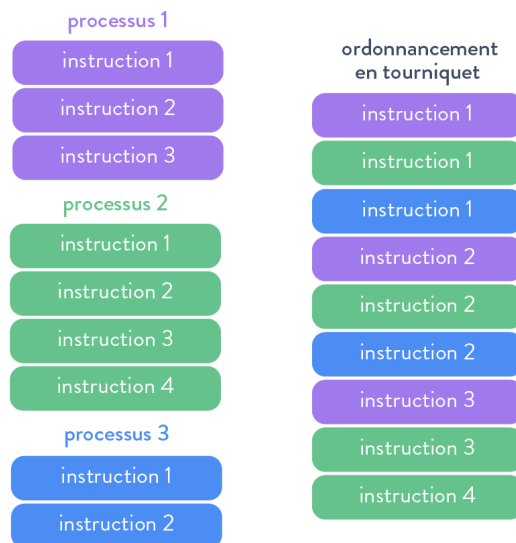
L'**ordonnancement** des processus consiste à organiser et contrôler les différents processus et leur bascule d'un état à un autre.

Les processus en cours d'exécution peuvent prendre trois états :

- **prêt** (le programme est paré à fonctionner et attend l'autorisation pour le faire) ;
- **actif** (le programme a été autorisé et utilise ponctuellement le processeur ou la ressource demandée) ;
- **bloqué** (le programme attend la survenue d'un événement externe pour pouvoir poursuivre son exécution).



Des processus peuvent lancer d'autres processus, ou nécessiter des échanges ou des synchronisations avec d'autres processus. Dans certains cas, plusieurs processus peuvent avoir besoin d'une même ressource au même moment.



© SCHOOLMOUV



À retenir

La gestion précise des processus est variable d'un système d'exploitation à l'autre, et peut comporter des notions de priorités, mais l'objectif reste identique :

- optimiser l'exécution des processus en minimisant les « temps morts » et en évitant les situations d'**interblocage**.



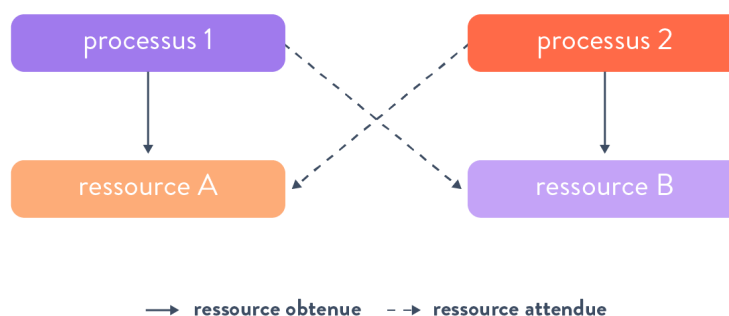
Définition

### Interblocage :

Un interblocage (*deadlock* en anglais) est un blocage réciproque. Ce problème survient quand deux processus attendent mutuellement après une ressource bloquée par l'autre.

Le processus P1 retient la ressource A dans l'attente de la ressource B.  
Le processus P2 retient la ressource B dans l'attente de la ressource A.

→ Ils se bloquent mutuellement.



© SCHOOLMOUV

En l'absence de mécanisme permettant au processus d'éviter cet état ou d'en changer, le blocage peut perdurer indéfiniment.

→ C'est pourquoi les systèmes d'exploitation intègrent des mécanismes de détection ou de prévention des interblocages.

On peut également rencontrer des mécanismes de protection contre l'interblocage à d'autres niveaux. C'est le cas pour le langage Python, dont l'interpréteur intègre un dispositif de verrouillage qui limite l'exécution à un seul fil (*thread* en anglais). Ce mécanisme est appelé GIL pour *Global Interpreter Lock*, soit « verrou global de l'interpréteur ». Notons cependant que ce dispositif affecte notablement les performances générales du système informatique.

## d. Interfaces utilisateur

Les systèmes d'exploitation proposent des **interfaces**, accessibles à l'utilisateur ou à l'administrateur de l'appareil. Les deux catégories d'interfaces sont :

- la ligne de commande (CLI en anglais, pour *Command Line Interface*) ;
- l'interface graphique (GUI en anglais, pour *Graphical User Interface*).

- 1 La **ligne de commande** est l'interface historique des systèmes d'exploitation. Elle nécessite de saisir des commandes, sous la forme de textes, souvent assorties de paramètres, dans un style abrégé et selon des conventions syntaxiques analogues aux langages de programmation. Bien qu'elle paraisse moins moderne que l'interface graphique, l'interface en ligne de commande offre de nombreuses possibilités et s'avère très efficace.



L'utilisation de la ligne de commande n'est pas à la portée des débutants.


- 2 L'**interface graphique** est l'interface par défaut pour l'utilisateur d'un ordinateur grand public moderne, d'une tablette ou d'un smartphone. Les interactions s'effectuent avec une souris ou au toucher sur un écran tactile. Pour les logiciels et les applications grand public, l'utilisateur est guidé et accompagné dans ses démarches.



L'intégration d'un système d'interface graphique au sein d'un système d'exploitation évite aux développeurs d'avoir à concevoir ces éléments d'interface (icônes, fenêtres, boutons, ascenseurs etc). Cela permet aussi de conserver un aspect identique d'un logiciel ou d'une application à l'autre, facilitant ainsi la prise en main et l'usage par les utilisateurs.

Conclusion :

Les appareils électroniques et informatiques modernes comportent un nombre parfois important de composants miniaturisés. Nous avons



d'abord présenté les architectures matérielles qui permettent l'intégration de ces composants dans différents types d'appareils, en particulier les micro-ordinateurs et les smartphones. Nous avons ensuite montré le rôle central du système d'exploitation de ces appareils, à la fois couche d'abstraction et chef d'orchestre permettant l'usage aisé de logiciels applicatifs, par une gestion appropriée des ressources matérielles.