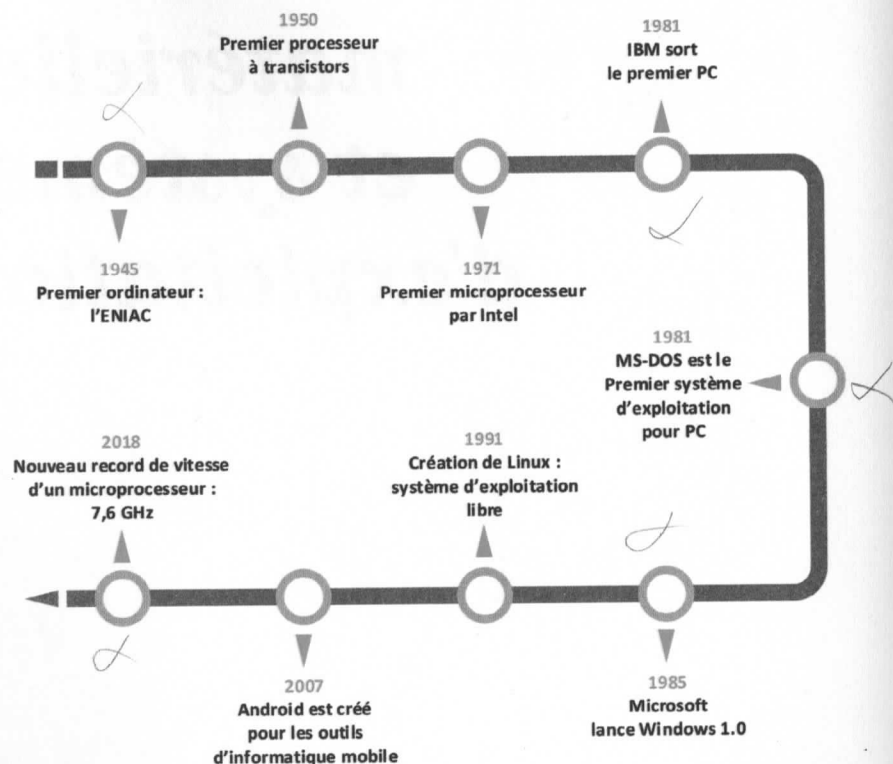


Cours

1 Un peu d'histoire

Du premier processeur, ouvrant la voie à l'informatique, aux récents microprocesseurs et aux systèmes d'exploitation mobiles, l'histoire a été rapide. Nous la retraçons ici en quelques dates clés :



2 Microprocesseur et mémoire

L'architecture de base

Le fonctionnement d'un outil numérique, quelle que soit sa nature (ordinateur, tablette, téléphone, assistant GPS, appareil photo...) est basé sur deux éléments fondamentaux :

- le **processeur**

Aussi appelé microprocesseur (car sa taille miniaturisée lui permet d'être intégré à n'importe quel élément numérique actuel), c'est lui qui est en charge d'effectuer les calculs élémentaires nécessaires à tout fonctionnement.

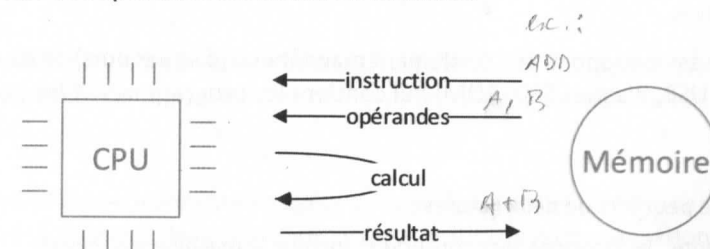
Le processeur porte aussi le nom de CPU (*Central Processing Unit*).

- la **mémoire**

Elle est chargée de stocker (de manière plus ou moins statique selon sa nature), les données nécessaires à toute opération.

En utilisant ces deux éléments, un principe de base permet l'activité numérique : tout programme est une suite d'opérations simples qui ont toutes la même forme :

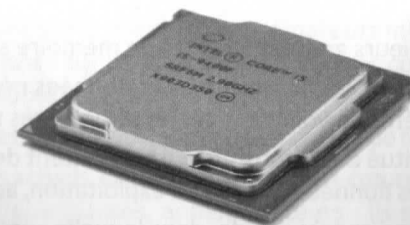
- 1 une instruction élémentaire à effectuer est chargée de la mémoire sur le processeur
- 2 les opérandes (données sur lesquelles va être fait le calcul) sont chargées de la mémoire sur le processeur
- 3 le calcul de l'opération élémentaire est effectué
- 4 le résultat de l'opération est stocké en mémoire



Les microprocesseurs

Au niveau technique, le microprocesseur est un circuit électronique intégré qui effectue des opérations. Sa taille est de plus en plus réduite.

Les opérations qu'est capable d'effectuer un microprocesseur sont son jeu d'instructions.



La vitesse d'un microprocesseur est définie par son horloge : l'horloge fournit le rythme des tâches élémentaires effectuées, en Hz (nombre de pulsations par seconde).

Concept

La rapidité à effectuer des instructions par un microprocesseur s'exprime en MIPS (Millions d'Instructions Par Seconde).

Historiquement, deux familles de microprocesseurs sont disponibles sur le marché, basées sur des fonctionnements opposés :

- les processeurs RISC (*reduced Instruction Set Computer*) proposent un nombre restreint d'instructions, qu'il est possible d'effectuer efficacement et très rapidement
- les processeurs CISC (*Complex Instruction Set Computer*) disposent d'un nombre d'instructions plus important et plus élaborées, mais sont donc moins rapides pour effectuer ces instructions

Le choix du processeur selon le besoin a donc une importance, mais notons que les dernières évolutions en termes de rapidité permettent de créer des RISC très puissants dont l'utilisation peut être comparée à celle des CISC, rendant la spécificité de chaque famille moins évidente.

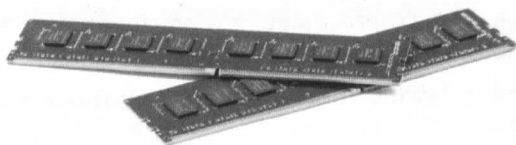
Les mémoires

La mémoire est le support, principalement magnétique (disques durs) ou électronique (RAM, Clés USB, disques SSD, ROM) qui contient les programmes et les données.

Concept

La mémoire peut être de deux natures :

- permanente : les données sont conservées lorsque la machine est éteinte
- volatile : les données ne sont conservées que pour la durée de leur utilisation



Dans un ordinateur, plusieurs grandes familles de mémoire sont utilisées :

- La mémoire vive contient les programmes et données nécessaires au microprocesseur. C'est une mémoire volatile, mais accessible très rapidement.
- Le stockage est constitué des dispositifs qui permettent de conserver de manière permanente toutes les données (système d'exploitation, applications, données...).
- Le cache est une mémoire très rapide, dans laquelle sont stockées des données auxquelles le microprocesseur a besoin d'accéder souvent, permettant ainsi un gain de temps.

- Le registre est une mémoire, de taille réduite, mais directement intégrée dans le microprocesseur, pour un gain de temps d'accès très important. Pour les microprocesseurs actuels, la taille du registre est un facteur important de choix et de prix.

3 Les systèmes sur puces : les SoCs**Le principe**

Rendu possible par la miniaturisation permanente des composants électroniques, l'idée de base est d'intégrer dans une seule puce plusieurs éléments, de natures et de rôles différents, pour créer un système autonome capable d'effectuer une tâche spécifique.

Ces dispositifs portent le nom de systèmes sur puce, ou *System On a Chip*, d'où l'appellation actuelle de SoCs.

Concept

Le principe élémentaire d'un système sur puce est d'effectuer une tâche définie, de manière complète, robuste et rapide. Ce sont ces objectifs qui définissent les éléments à intégrer dans le SoC.

Les éléments qui peuvent être intégrés dans un SoC sont extrêmement nombreux : des microprocesseurs, de la mémoire, des dispositifs de communication sans fil, des dispositifs d'entrée/sortie, des capteurs...

Exemples

Nous prenons ici quelques exemples de natures différentes pour illustrer l'apport des SoCs dans les appareils numériques actuels.

- 1 Certains appareils de **photo numérique** intègrent des SoCs très complets de traitements d'image, offrant des fonctions qui devaient auparavant être réalisées par des applications logicielles.
- 2 Des fabricants ont mis sur le marché des **ordinateurs miniaturisés** complets, dits nano-ordinateurs, basés sur l'utilisation de SoCs très complets, voire sur un seul SoC qui regroupe toutes les fonctionnalités d'un ordinateur habituel.
- 3 Dans les **smartphones**, un seul SoC peut être en charge de toutes les communications.
- 4 Les microprocesseurs actuels, dits **processeurs multi-cœurs**, sont équipés de plusieurs unités de travail indépendantes, appelées cœurs : ils sont capables d'effectuer plusieurs opérations simultanément, c'est-à-dire qu'il est devenu possible d'effectuer un nombre plus élevé d'instructions que l'horloge ne donne de pulsations.

Des avantages nombreux

Les avantages des systèmes sur puce sont nombreux :

- La **vitesse de traitement** et donc l'efficacité sont accrues. En effet, la proximité des composants sur le circuit électronique miniaturisé réduit les distances, l'utilisation d'éléments moins génériques améliore l'efficacité.
- Le regroupement des éléments ne nécessite plus d'alimentations multiples, la **consommation énergétique** est réduite de manière significative, entraînant une baisse du coût d'énergie et une amélioration de la gestion de l'énergie (principalement pour l'informatique mobile et les smartphones, pour lesquels la gestion des performances de la batterie est importante).
- Même si les coûts d'ingénierie sont plus élevés sur la phase de conception, les **coûts de matières premières et de fabrication** sont eux aussi réduits par rapport à une architecture classique.
- Les possibilités de **miniaturisation** des ordinateurs et de tous les autres équipements numériques sont encore accrues, offrant ainsi un fort potentiel d'innovation.
- L'**adaptation au besoin** est affinée : la spécificité de chaque système permet une efficacité optimale.

4 Les systèmes d'exploitation

Le fonctionnement général

Le système d'exploitation est un ensemble de programmes qui va permettre d'utiliser les éléments physiques d'un ordinateur pour exécuter les applications nécessaires à l'utilisateur.

L'élément fondamental du système d'exploitation est le noyau, c'est lui qui permet et gère l'accès aux ressources matérielles.

Ses principales fonctions sont :

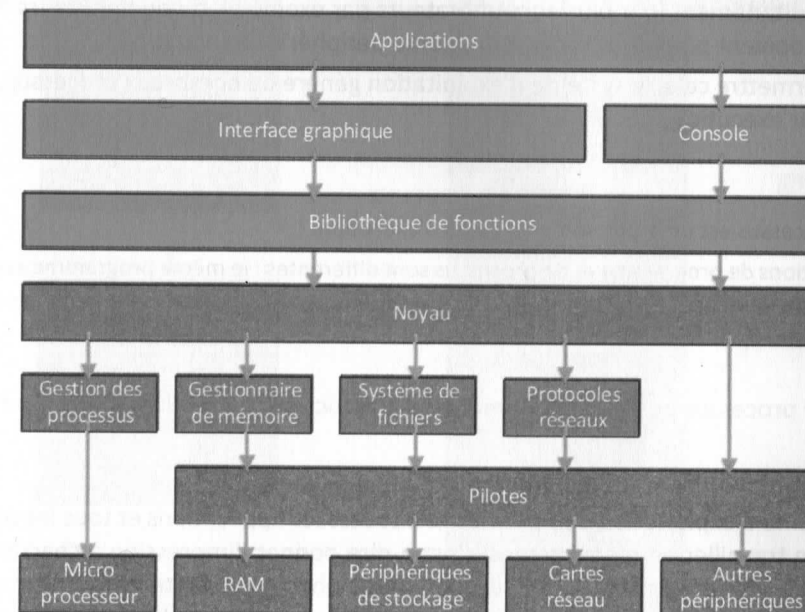
- le dialogue avec les périphériques (microprocesseur, mémoire, disques, cartes graphique, carte réseau, clavier, souris...)
- l'exécution par le microprocesseur des programmes souhaités par les utilisateurs et l'ordonnancement de ces tâches
- la gestion des accès aux ressources, pour permettre d'une part à tous les utilisateurs de travailler simultanément, et d'autre part de ne permettre l'utilisation d'une ressource qu'aux utilisateurs autorisés.

Au-dessus du noyau, de très nombreux programmes sont en charge de toutes les fonctions qui sont offertes aux programmes utilisateurs pour permettre une utilisation complète et optimale de la machine physique (gestionnaire de fichiers, lecture de sons, gestion de l'énergie, gestion des communications réseau, gestion des performances...)

Les systèmes d'exploitation actuels proposent aussi de nombreux outils de niveau supérieur, qui apportent du confort de travail à l'utilisateur, jusqu'à lui éviter l'installation de programmes à part entière (navigateur Internet, outils de traitement d'image, logiciel de messagerie, traitement de texte, outils de diagnostic...).

Les différents éléments

Le schéma suivant illustre l'architecture globale d'un système d'exploitation :



Au niveau utilisateur, nous trouvons les **applications**, exécutées via l'**interface graphique** ou directement en **mode commandes**. Les applications peuvent utiliser des **bibliothèques de fonctions**.

Ces applications s'appuient sur le **noyau**, élément central du système d'exploitation, qui génère des appels système pour accéder à une ressource.

Selon la nature de la ressource nécessaire, des gestionnaires spécifiques sont sollicités : le **gestionnaire de processus** pour l'exécution d'un programme par le microprocesseur, le **gestionnaire de mémoire** pour l'accès à une donnée en mémoire, le **système de fichiers** pour la gestion des périphériques de stockage de masse (disque dur, DVD...), les **protocoles réseaux** pour les outils de gestion des différents réseaux disponibles.

Chaque **ressource physique** est gérée par un **pilote**, seule entité logicielle capable du dialogue avec le périphérique.

Pour de nombreux périphériques, un gestionnaire spécifique n'est pas nécessaire : le noyau peut solliciter directement le pilote concerné.

"thread", PIP, PPIP, "task"

5 Les processus

Objectifs

Pour permettre le fonctionnement d'un ordinateur, de nombreuses tâches ou applications doivent être exécutées simultanément, par le système d'exploitation et les différents utilisateurs.

Notons aussi qu'une même application (programme) doit pouvoir s'exécuter plusieurs fois simultanément (par plusieurs utilisateurs par exemple), ou que plusieurs applications doivent pouvoir accéder à un même périphérique sans conflit.

Pour permettre cela, le système d'exploitation génère de nombreux processus, puis gère leur exécution.

Concept

Un processus est un programme en cours d'exécution.

Les notions de programme et de processus sont différentes : le même programme exécuté plusieurs fois (dans le temps ou par plusieurs utilisateurs simultanément) générera plusieurs processus.

Chaque processus possède en mémoire les instructions à exécuter et ses données.

L'ordonnancement

Le système d'exploitation doit permettre à toutes les applications et tous les utilisateurs de travailler en même temps, c'est-à-dire donner l'impression à chacun qu'il est seul à utiliser l'ordinateur et ses ressources physiques. Cette gestion complexe des processus est réalisée par une partie spécifique du noyau : l'ordonnanceur.

Concept

Comme une ressource (le processeur ou un périphérique) ne peut pas être partagée, c'est son temps d'utilisation qui va l'être : le temps d'utilisation d'une ressource est partagé en intervalles très courts, pendant lesquels l'ordonnanceur l'alloue à un seul utilisateur.

L'ordonnanceur permet :

- de minimiser le temps de traitement du processus d'un utilisateur
- de garantir l'équité entre les différents utilisateurs
- d'optimiser l'utilisation de la ressource
- d'éviter les blocages.

Plusieurs algorithmes d'ordonnancement sont possibles, parmi les plus répandus, nous pouvons citer :

- Le **tourniquet** : la ressource est affectée à chaque processus à tour de rôle. Pour l'exécution simultanée des processus, c'est la rapidité de ce tour de rôle qui va donner l'impression à chaque utilisateur que son processus est seul à utiliser le processeur. Cette méthode ancienne a les avantages de sa simplicité, de sa rapidité de gestion et de sa robustesse.

Processus 1

instruction 1
instruction 2
...
...
...
instruction z

Processus 2

instruction 1
instruction 2
...
...
...
instruction y

Processus 3

instruction 1
instruction 2
...
...
instruction t

Exécution par le microprocesseur

instruction 1
instruction 1
instruction 1
instruction 2
instruction 2
instruction 2
...
...
...
...
...
...
instruction t
...
instruction y
...
instruction z

- La mise en place d'un système de **priorités** : l'ordre d'affectation de la ressource sera alors fonction de la priorité de la tâche. Cette méthode est très équitable, mais la définition du niveau de priorité d'une tâche doit être objective.
- La gestion du **premier entré, premier sorti** (FIFO : *First In, First Out*). L'exemple le plus évident de cet algorithme est la file d'impression des documents sur une imprimante.
- L'algorithme du « **plus court d'abord** » : très efficace pour satisfaire au mieux les utilisateurs, mais il n'est pas toujours simple d'évaluer le temps d'exécution d'une tâche avant son début.

Parallèlement à l'évolution des performances des microprocesseurs, l'ordonnancement est aussi un moyen d'amélioration de la rapidité de traitement : des algorithmes récents, de plus en plus complexes, ont été proposés récemment.

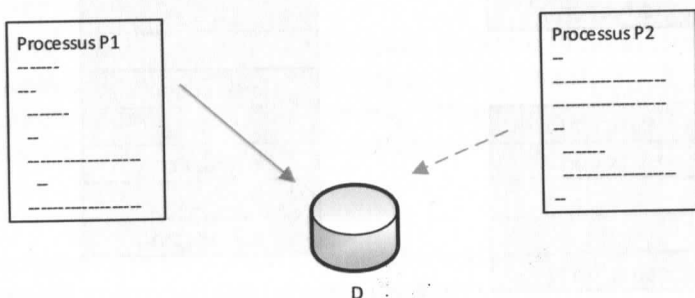
Les interblocages

Nous avons dit précédemment que des processus peuvent avoir besoin de la même ressource.

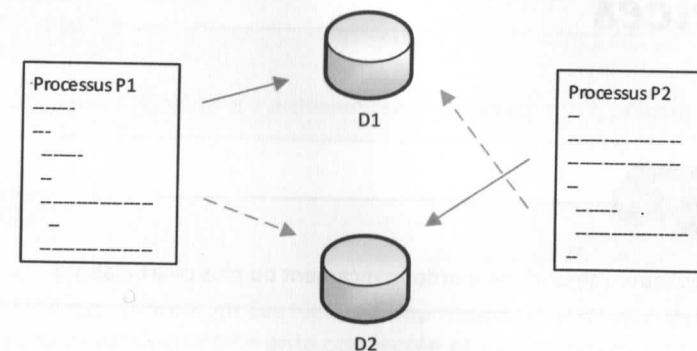
Dans de nombreuses situations, deux processus (ou davantage) peuvent souhaiter accéder à la même donnée sur le disque dur :

- Les deux processus ont uniquement besoin de lire la donnée : celle-ci est alors partagée, sans problème complexe.
- Les deux processus ont besoin de la donnée de manière exclusive, pour la modifier par exemple.
- Les deux processus ont besoin de communiquer entre eux : l'un doit attendre un résultat de l'autre.

Sur le schéma suivant, les processus P1 et P2 ont tous les deux besoins de la même donnée D pour la modifier, c'est-à-dire de manière exclusive. Le premier à y accéder est P1, D lui est allouée par le système d'exploitation. Lorsque P2 souhaite accéder à D, la ressource n'est pas disponible : P2 est alors bloqué jusqu'à la fin de l'utilisation de D par P1.



Prenons maintenant un autre exemple : deux processus P1 et P2 ont tous les deux besoins de deux données, nommées sur le schéma suivant D1 et D2. Voici une situation qui peut se produire :



Chaque processus bloque une donnée et est en attente de l'autre, rien ne pourra évoluer sans une intervention extérieure : cette situation porte le nom d'interblocage.

Face à cette problématique, deux solutions sont envisageables :

- essayer d'éviter un interblocage
- détecter qu'un interblocage est apparu, et le supprimer

La plupart des systèmes d'exploitation ont choisi de ne pas essayer d'éviter les interblocages, mais de les détecter s'ils surviennent et de les solutionner.

Nous n'irons pas plus en avant ici sur ces techniques, qui ne relèvent pas du programme de cette matière.