

Systeme d'exploitation

Qu'est ce qu'un OS ?

Nous attendons de nos ordinateurs qu'ils soient toujours plus performants, simples d'utilisation, capables d'accueillir des applications de plus en plus sophistiquées et des données toujours plus volumineuses.

Le hardware ne peut à lui seul répondre à ces exigences. Il faut donc un système capable d'orchestrer le matériel et les applications tout en nous donnant l'impression que l'ensemble est convivial et simple.

Le système d'exploitation

Le **système d'exploitation des ressources de l'ordinateur** communément appelé "système d'exploitation" ou *Operating System O.S.* est généralement représenté comme une couche logicielle placée au dessus du hardware.

L'interface utilisateur

Au système d'exploitation est associée une interface utilisateur (texte ou graphique) capable d'interpréter une série de commandes de base. Cette interface ne fait (théoriquement) pas partie du système d'exploitation mais l'étude de ses commandes donne un bon aperçu des fonctions que procure l'OS.

La convivialité du système d'exploitation dépend de la richesse et de la simplicité de l'interface homme/machine. Les deux sont donc étroitement liés.

L'apport du système d'exploitation aux applications

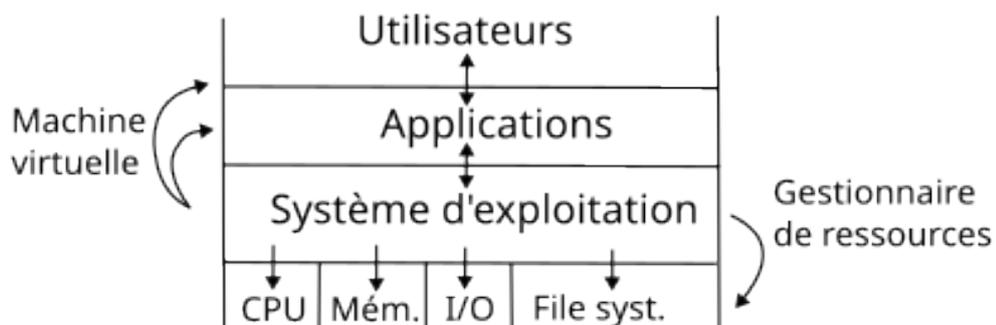
Le système d'exploitation fonctionne comme une **machine virtuelle** qui, pour les applications, substitue des composants logiciels aux composants matériels.

Du point de vue des développeurs, il est la plateforme pour laquelle sont construites les applications.

Pour ce qui est du développement des applications : l'OS propose une interface de programmation appelée **API (Application Program Interface)**, une sorte de boîte à outils à laquelle les développeurs recourent pour construire leurs applications.

Ces API procurent une vue uniforme et simplifiée des ressources de la machine.

Cela permet aux applications de faire abstraction des particularités du matériel en dissimulant la diversité et la complexité du hardware.



Rôle du système d'exploitation vis-à-vis des ressources système

Le S.E. reçoit des demandes des programmes d'application et des utilisateurs. Il leur donne suite en leur allouant les ressources du système :

- 1° le CPU
- 2° la mémoire
- 3° les périphériques
- 4° le système de fichier (y compris parfois, le réseau)

L'allocation des ressources système est une tâche complexe :

- Répartir les temps d'utilisation du CPU entre chaque application, tenir compte des priorités, des applications en attente etc.
- Partager la mémoire entre les différentes applications (mémoire physique ou virtuelle)
- Partager les accès aux périphériques : disques, carte graphique, clavier, souris etc.
- Donner le moyen d'enregistrer des données sur les disques.
- Rediriger certaines commandes du système de fichiers (*file system*) vers le réseau
- Identifier les utilisateurs et gérer les droits d'accès aux fichiers et aux périphériques
- Gérer les erreurs



La référence à ce sujet :

L'ouvrage « *Systèmes d'exploitation* » de Andrew Tanenbaum, informaticien et pédagogue hors pair et dont voici le site personnel <http://www.cs.vu.nl/~ast/>



A voir aussi : la présentation power point accessible à l'adresse.

http://www.prenhall.com/divisions/esm/app/author_tanenbaum/custom/powerpoint.html

Terminologie:

Un **programme** est une suite statique d'instructions.

Un **processeur** est l'agent qui exécute les instructions d'un programme

Un **processus** est un programme ou une fraction de programme en cours d'exécution. Un programme peut avoir plusieurs exécutions simultanées. Pour faire tourner un processus il faut donc, non seulement chercher le code et les données mais il faut aussi lui réserver un espace mémoire, gérer les accès aux ressources, la sécurité etc. C'est un des rôles de l'OS.

Monotâche

Pour d'anciens systèmes d'anciens systèmes d'exploitation, comme le DOS par exemple, le rôle du système d'exploitation se limitait à charger les programmes d'application en mémoire avant de leur passer la main.

À présent, les systèmes d'exploitation sont multitâches. Le temps processeur est partagé par plusieurs processus qui semblent tourner simultanément.

Multitâche

Le système d'exploitation est multitâche s'il permet à l'ordinateur de faire tourner simultanément plusieurs processus. Il s'arrange pour que chaque processus ait à tour de rôle un court laps de temps pour s'exécuter partiellement. Cette méthode, aussi appelée **multiprogrammation**, donne l'impression que tous ces programmes s'exécutent simultanément.

L'un des principaux avantages du multitâche est que cela maximise l'occupation du processeur. Dès qu'un programme s'interrompt pour attendre la réalisation d'une fonction plus lente, telle qu'une opération d'entrées/sortie, le processeur passe sans plus attendre à l'exécution d'un autre processus.

Multitâche coopératif Anciennement (Windows 95 - 98)

Chaque processus dispose des ressources du système puis les libère pour permettre au processus qui suit dans une file d'attente d'en disposer à son tour.

⇒ Chaque tâche dépend des autres. Une tâche programmée de manière peu rigoureuse va compromettre la bonne marche de l'ensemble.

Multitâche préemptif Les systèmes d'exploitation actuels

Chaque processus dispose du processeur pendant un laps de temps déterminé à l'avance ou jusqu'à ce qu'un autre ait une priorité supérieure à celle du processus en cours.

L'ordonnancement, (l'attribution du temps processeur aux processus) est fait par l'OS sans que les applications soient consultées.

⇒ Une application qui "se plante" est mise de côté et ne bloque plus le système.

Multithreading

Une application multithread est un logiciel qui dès sa conception a été partagé en différentes unités d'exécutions ou threads dont la gestion est confiée au système d'exploitation. Exemple : la correction orthographique dans WORD.

Nous avons vu que le multitâche consiste à scinder les applications en plusieurs processus. Ces processus sont indépendants les uns des autres en ce sens qu'ils occupent chacun une partie distincte de la mémoire.

Le multithreading est une technique un peu semblable mais au lieu de subdiviser les applications en processus indépendants, l'application est subdivisée en ce qu'on appelle des *fils d'exécution* ou *threads* ou encore « processus légers ».

Contrairement aux processus, les threads d'une application partagent le même espace d'adressage. Les processeurs actuels intègrent dans leur architecture des dispositifs qui mettent cette technique du multithreading à profit.