

## A - PRINCIPES DE BASES DES SYSTEMES D'EXPLOITATION

Le système d'exploitation d'un ordinateur est un programme (couche logicielle) qui assure toutes les tâches relevant de deux aspects. D'une part, il facilite l'accès aux ressources de la machine tout en assurant une certaine portabilité des développements et, d'autre part, il contrôle que tous ces accès sont licites en protégeant ses propres données, celles des utilisateurs et celles de toutes les tâches s'exécutant sur la machine. Ce programme est stocké pour partie en mémoire centrale pour les fonctions les plus couramment utilisées, et en partie sur le disque dur.

Il gère les échanges entre le microprocesseur, la mémoire centrale et les unités périphériques écran, souris, clavier, disque dur etc...), les différentes interfaces parallèles ou séries, etc...Le SE met à la disposition de l'utilisateur un certain nombre de commandes dont il assure l'exécution, comme le formatage de disquettes, la copie de fichiers, ou le paramétrage de certains matériels. Il détecte les erreurs, en informe l'utilisateur et prend en compte les choix ou corrections apportés par ce dernier.

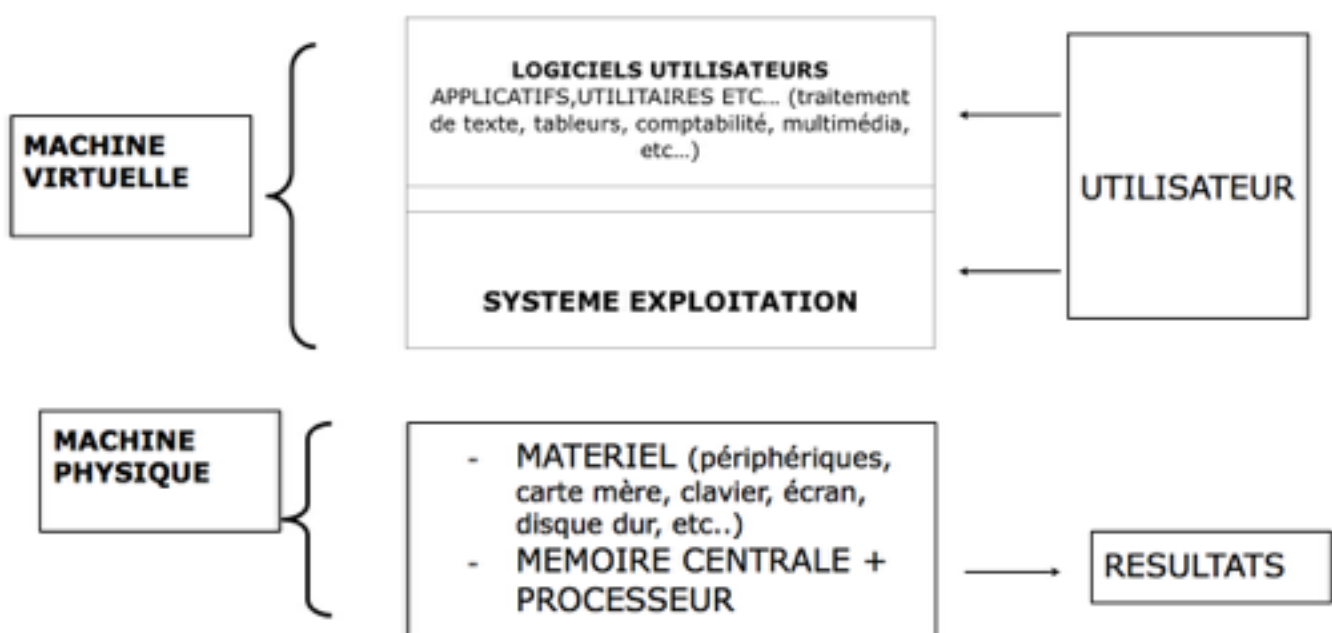
On le nomme aussi OS (operating system) , ou SE (système d'exploitation en français).

### Définition :

#### Le système d'exploitation est une machine virtuelle

Le système d'exploitation est une machine virtuelle plus simple à programmer que la machine réelle, physique. Il offre une interface de programmation à l'utilisateur qui n'a donc pas besoin de connaître le fonctionnement réel de la machine : l'utilisateur demande au système d'effectuer certaines tâches et le système se charge ensuite de dialoguer avec la machine pour réaliser les tâches qui lui sont demandées. L'OS gère l'ensemble des transactions de l'ordinateur indépendamment des applications utilisées.

Dans cette optique, la machine réelle est en fait cachée sous le système d'exploitation et ce dernier permet donc un dialogue abstrait entre la machine et l'utilisateur. Par exemple, l'utilisateur se contente de dire « je veux écrire mes données dans un fichier sur le disque dur » sans se préoccuper de l'endroit exact où se trouvent ces données, ni de la façon dont il faut s'y prendre pour les copier sur le disque dur.



Notons qu'à cet effet le système d'exploitation se doit de proposer une interface permettant ce dialogue abstrait et c'est pour cela que des concepts dénués de toute réalité physique sont généralement utilisés lorsqu'on parle des systèmes d'exploitation. Citons par exemple les variables qui permettent de faire référence à des valeurs stockées quelque part en mémoire, **les fichiers** qui représentent l'abstraction du stockage de données ou **les processus** qui sont l'abstraction d'un programme en cours d'exécution.

En simplifié un système d'exploitation a de nombreuses responsabilités :

- Permettre l'exécution et le contrôle des logiciels applicatifs.
- Gérer les périphériques matériels (processeur, mémoire vive, carte graphique...).
- Fournir une Interface Homme-Machine (IHM) à un ou plusieurs utilisateurs.

Pour reprendre l'analogie automobile, nous dirions que le système d'exploitation en tant que machine virtuelle est ce qui relie les éléments mécaniques de la voiture au conducteur. Il propose une interface plus ou moins standard (le volant, les pédales d'accélérateur, de frein et d'embrayage, etc.) qui permet au conducteur de dialoguer avec le moteur et d'obtenir un travail de ce dernier, sans pour autant se soucier du fonctionnement réel de ce moteur.

### L'Interface **Homme-Machine**

L'IHM d'un système d'exploitation permet à l'utilisateur d'interagir avec le système : lancement de commandes, utilisation de logiciels, accès aux fichiers...

Jusqu'au milieu des années 1980, les systèmes disposaient uniquement d'une IHM textuelle.

De nos jours, la plupart des systèmes offrent une IHM graphique conviviale, ou GUI (Graphical User Interface). Il reste possible de les utiliser en mode texte grâce à un terminal. Elle présente à l'utilisateur un bureau virtuel et permet le pilotage du système à la souris.

### **Le système d'exploitation est un gestionnaire de ressources**

Les ressources d'une machine sont tous les composants qui sont utilisés pour effectuer un travail et qu'un utilisateur de la machine pourrait s'approprier. À ce titre, tous les périphériques comme la mémoire ou les disques durs sont des ressources. Les registres du processeur ou le temps passé par le processeur à faire des calculs sont aussi des ressources.

Le système d'exploitation est **un gestionnaire de ressources**, c'est-à-dire qu'il contrôle l'accès à toutes les ressources de la machine, l'attribution de ces ressources aux différents utilisateurs de la machine et la libération de ces ressources quand elles ne sont plus utilisées.

Ce contrôle est capital lorsque le système permet l'exécution de plusieurs programmes en même temps ou l'utilisation de la machine par plusieurs utilisateurs à la fois. En particulier, il doit veiller à ce qu'un utilisateur ne puisse pas effacer les fichiers d'un autre utilisateur ou à ce qu'un programme en cours d'exécution ne détruise pas les données d'un autre programme stockées en mémoire.

Un autre aspect capital du contrôle des ressources est la gestion des conflits qui peuvent se produire quand plusieurs programmes souhaitent accéder en même temps aux mêmes données. Supposons par exemple qu'un utilisateur exécute deux programmes, chacun d'eux écrivant dans le même fichier. Il y a de grande chance que, si aucune précaution n'est prise, le résultat contenu dans le fichier ne soit pas celui escompté.

Si nous nous référons à notre analogie automobile, le système d'exploitation en tant que gestionnaire de ressources est représenté par le limiteur de vitesse qui oblige le conducteur à conduire sagement ou par le système d'arrêt automatique en cas d'endormissement qui empêche le conducteur endormi d'entrer en collision avec un mur.

### **Le système d'exploitation intervient peu**

Afin de contrôler que l'accès à toutes les ressources de la machine s'effectue dans de bonnes conditions, que ce soit pour l'allocation de mémoire ou de temps CPU, l'écriture de données, la lecture de données ou la libération d'une ressource, nous pourrions envisager de faire appel au système d'exploitation pour chaque requête d'accès. Ainsi un programme qui souhaite écrire en mémoire pour, par exemple, affecter une valeur Y à la variable X devrait faire appel au système. Si ce programme effectue une boucle de 10 000 affectations, il faudrait alors faire 10 000 fois appel au système pour contrôler l'accès à la mémoire.

Bien entendu, ce n'est pas raisonnable car cela va à l'encontre de la dualité efficacité contre commodité. Une des clés pour un système d'exploitation réussi réside donc dans sa capacité à contrôler tous les accès aux ressources en intervenant le moins souvent possible de façon dynamique .

Pour le moment, il faut garder à l'esprit qu'il existe des moyens pour que le système d'exploitation contrôle les accès aux ressources sans pour autant pénaliser le fonctionnement de la machine. La MMU est un de ces moyens : elle contrôle l'accès aux différentes zones de mémoire à partir de tables de pages et le système a juste besoin d'écrire de temps en temps dans ces tables pour assurer en permanence la protection de la mémoire entre les différentes tâches.

### Mode utilisateur contre mode noyau

Les processeurs fonctionnent dans deux modes différents : le mode noyau où toutes les instructions sont autorisées et le mode utilisateur où certaines instructions sont interdites.

Le système d'exploitation peut utiliser cette propriété pour faciliter le contrôle qu'il exerce :

il s'exécute en mode noyau alors que tous les autres programmes sont exécutés en mode utilisateur. Ces programmes utilisateur ont ainsi par essence des pouvoirs limités et certaines opérations leurs sont interdites.

Par exemple, en n'autorisant l'accès aux différentes ressources de la machine qu'aux programmes s'exécutant en mode noyau, le système d'exploitation protège ces ressources et contraint les programmes utilisateur à faire appel à lui (pas nécessairement de façon dynamique) pour accéder aux ressources de la machine.

### Les appels système

Certaines opérations ne peuvent pas être contrôlées de façon statique par le système d'exploitation et un programme utilisateur doit donc faire explicitement appel aux services du système d'exploitation pour accéder à certaines ressources. À ces fins, le système d'exploitation propose une interface de programmation, c'est-à-dire qu'il permet d'accéder à un certain nombre de fonctionnalités qu'il exécutera pour l'utilisateur. Les appels système sont l'interface proposée par le système d'exploitation pour accéder aux différentes ressources de la machine.

Par exemple, il est nécessaire de faire appel au système d'exploitation pour créer un fichier sur le disque dur et cela via un appel système. Si nous demandons que ce fichier soit créé dans un répertoire qui nous est interdit d'accès, par exemple un répertoire appartenant à un autre utilisateur, l'appel système va refuser de créer le fichier.

### Des appels système très nombreux

Le nombre d'appels système proposés varie suivant les systèmes d'exploitation et, pour un même type de système d'exploitation ce nombre diffère suivant les versions utilisées.

Ces appels reflètent les services que peut rendre le système d'exploitation et il sont généralement classés en quatre catégories :

- gestion des processus ;
- gestion des fichiers ;
- communication et stockage d'informations ;
- gestion des périphériques.

Notons au passage que les trois premières catégories correspondent à trois concepts sans réalité physique...

### Le cas d'UNIX

Sous UNIX, les périphériques sont gérés comme de **simples fichiers** - ce qui représente l'abstraction absolue - et il n'y a donc que trois catégories d'appels système. Le manuel en ligne **man** permet de connaître la liste des appels système et le but de chaque appel : tous les appels système se trouvent dans la section 2 du manuel.

Les appels système peuvent être directement utilisés dans un programme C et il faut savoir que la plupart des fonctions de la bibliothèque standard utilisent aussi ces appels système. La deuxième partie de ce document donnera de nombreux exemples d'utilisation des appels système.

### Exécution d'un appel système

Un appel système provoque en fait une interruption logicielle et il suffit alors de programmer la machine pour que la routine correspondant à l'interruption fasse partie du système d'exploitation.

En pratique, il est fréquent que toutes les interruptions aient la même routine associée : le système d'exploitation. Suivant le type d'interruption (matérielle ou logicielle) et suivant la

nature de l'interruption (erreur, travail terminé, etc.), le système décide alors quelle action il doit entreprendre.

En particulier, on peut se demander ce qui se passe lorsqu'une demande d'interruption matérielle a lieu au milieu d'une interruption logicielle. La réponse à cette question dépend non seulement de chaque système d'exploitation, mais aussi de l'implantation d'un même système sur plusieurs architectures différentes.

Les interruptions peuvent être synchrones ou asynchrones. Cela veut donc dire qu'un appel système peut donner lieu à une requête d'interruption qui sera traitée de façon asynchrone et que, par exemple, il se peut que le système ait d'autres choses plus importantes à faire. Néanmoins, du point de vue du programme qui exécute l'appel système, celui-ci se déroule de façon synchrone, c'est-à-dire que l'appel système interrompt l'exécution du programme et celui-ci ne peut pas reprendre son exécution tant que l'appel système n'est pas terminé.

Toutefois, et afin de ne pas trop pénaliser les programmes effectuant des appels système, le système d'exploitation utilise généralement des intermédiaires rapides et il rend la main au programme avant d'avoir effectivement accompli l'action demandée. Par exemple, si un programme demande l'écriture de données dans un fichier sur disque dur, le système d'exploitation va copier ces données dans un tampon qui lui appartient et va ensuite rendre la main au programme. Le système d'exploitation écrira ces données sur le disque dur à un autre moment.

Retenons les faits suivants :

- un appel système permet de demander au système d'exploitation d'effectuer certaines actions ;
- un appel système interrompt le programme qui l'exécute ;
- un appel système est exécuté en mode noyau même si le programme ayant demandé son exécution est exécuté en mode utilisateur.

L'utilisation des appels système illustre aussi un autre phénomène qui permet de développer des systèmes d'exploitation efficaces et cohérents : le meilleur moyen pour faire intervenir le système d'exploitation est de déclencher une interruption !

## Structure des systèmes d'exploitation

### Une représentation abstraite

La relation entre la machine, le système d'exploitation et les programmes des utilisateurs est toujours représentée sous forme d'empilement d'un certain nombre de couches aux frontières bien définies : le système d'exploitation reliant la machine et l'utilisateur, il s'intercale naturellement entre les couches représentant ces derniers.



Ces représentations sont abstraites, c'est-à-dire qu'elles ne correspondent pas à la réalité physique des choses : comme nous le disions plus haut, le système d'exploitation peut effectuer une partie de son travail de façon statique, c'est-à-dire sans qu'un programme utilisateur ait besoin de faire appel à lui.

cette représentation par couche (logicielle, matérielle) permet néanmoins de bien mettre en évidence les relations entre les différents acteurs et nous l'utiliserons tout en gardant bien à l'esprit que cela ne correspond pas nécessairement à la réalité.

On parle de couche d'abstraction : plutôt que se s'adresser directement aux différents périphériques, les logiciels passent par le système d'exploitation, qui traduit leurs appels auprès du matériel.

### Le noyau

Le noyau (ou kernel) est la partie fondamentale du système d'exploitation. Il est chargé en mémoire vive durant le démarrage de l'ordinateur (boot sequence).

Après le démarrage, la mémoire est divisée en deux parties :

- L'espace noyau, réservé au noyau lui-même.
- L'espace utilisateur, dédié aux applications.

Cette division permet d'augmenter la robustesse du système : aucun programme de l'espace utilisateur ne peut accéder à la mémoire du noyau.

Le noyau remplit les fonctions essentielles du système d'exploitation :

- Gestion du matériel.
- Exécution des programmes (sous la forme de processus).
- Echanges entre matériel et logiciels.

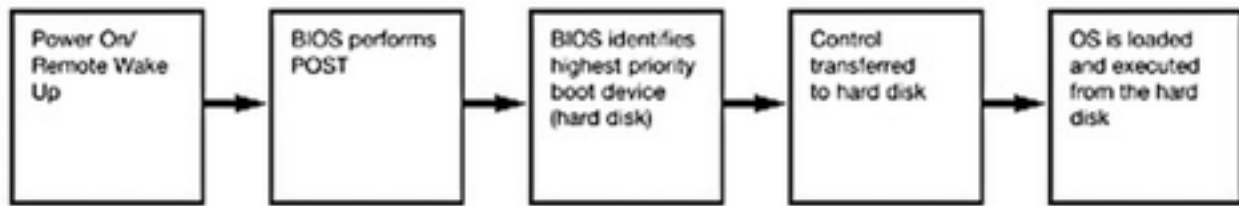


Figure 1. Normal PC Boot Sequence

## Les systèmes monolithiques

Le système d'exploitation d'une machine n'est en pratique pas constitué d'un seul programme. Il se compose de deux parties que l'on nomme souvent le noyau (*kernel* en anglais), qui est le cœur du système (*core* en anglais), et les programmes système. Le noyau est exécuté en mode noyau et il se peut qu'il corresponde à plusieurs processus différents (2 ou 3 sous les systèmes UNIX, par exemple).

Même si le noyau effectue l'essentiel des tâches du système d'exploitation, ce dernier ne peut se réduire à son noyau : la plupart des services qu'un utilisateur utilise sur une machine sont en fait des sur-couches du noyau. Les programmes système qui assurent ces services sont néanmoins exécutés en mode utilisateur et, donc, ils doivent faire appel au noyau comme n'importe quel utilisateur pour accéder aux ressources.

En pratique, il est toujours difficile de distinguer où finit le système d'exploitation et où commence l'environnement utilisateur. Formulé différemment, il est toujours difficile de dire si un programme s'exécutant en mode utilisateur fait partie du système d'exploitation ou pas. En revanche, la limite entre le noyau et les programmes système est claire : le noyau est toujours exécuté en mode noyau !

L'interface proposée par le noyau est l'ensemble des appels système et, comme tout programme s'exécutant en mode utilisateur, les programmes système utilisent ces appels. Il est donc traditionnel d'intercaler, dans notre modèle à couches, les appels système entre le noyau et les programmes système. De même et pour les mêmes raisons, il est traditionnel d'intercaler le jeu d'instructions entre le matériel et le noyau.

Les systèmes d'exploitation ont été développés petit à petit. Bien souvent, les développeurs de tels systèmes reprenaient les travaux menés par les développeurs du système précédent et les complétaient. De génération en génération, le noyau des systèmes d'exploitation se mit donc à grossir et cela mena vite à des systèmes difficilement exploitables.

Ainsi, n'importe quelle procédure du noyau peut en appeler n'importe quelle autre et l'implantation de nouveaux services est très délicate : une erreur à un endroit du noyau peut entraîner un dysfonctionnement à un autre endroit qui, a priori, n'a rien à voir.

Ces inconvénients entraînèrent le développement de systèmes d'exploitation fondés sur d'autres modèles.

## Les systèmes à couches

Les systèmes à couches appliquent au noyau le principe des couches tel que nous l'avons expliqué ci-dessus. L'idée consiste à charger chaque couche du noyau d'une tâche bien précise et à proposer une interface pour les couches au-dessus.

Chaque couche masque ainsi les couches situées en dessous d'elle et oblige la couche située au-dessus à utiliser l'interface qu'elle propose. L'avantage est évident : les couches sont totalement indépendantes les unes des autres et seule l'interface entre chaque couche compte. Cela permet de développer, de débiter et tester chaque couche en se fondant sur les couches inférieures qui sont sûres.

Même si ce principe est séduisant, il est très complexe à mettre en œuvre et les systèmes ainsi structurés sont souvent très lourds et peu performants.

## Les micro-noyaux

Les noyaux des systèmes monolithiques ont tendance à être volumineux. Par ailleurs, si un service particulier est très rarement utilisé, il doit néanmoins être présent dans le programme du noyau et il utilisera donc de la mémoire de façon inutile.

Pour éviter ces problèmes, une solution consiste à réduire le noyau à quelques procédures essentielles et à reporter tous les autres services dans des programmes système. Le noyau ainsi réduit s'appelle souvent micro-noyau.

Ce travail est néanmoins considérable car il suppose une refonte totale des systèmes d'exploitation habituels : ceux-ci ayant été développés au fil des ans, il se peut que des services très importants, mais apparus tardivement, soient implantés à la périphérie du noyau et que, donc, l'isolement de ces services demande la réécriture complète du noyau.

Cela permet néanmoins de clarifier les choses et de concevoir des systèmes très généraux se fondant uniquement sur des services essentiels, comme le partage du temps. La plupart des systèmes à micro-noyau en profitent aussi pour inclure des services pour les machines multiprocesseurs ou pour les réseaux de processeurs.

## Les systèmes mixtes

Certains systèmes monolithiques tentent d'alléger leurs noyaux en adoptant quelques principes des systèmes à couches ou des micro-noyaux. Par exemple, le système d'exploitation Linux permet de charger de façon dynamique des modules particuliers qui sont agglomérés au noyau lorsque celui-ci en a besoin.

Ce procédé est assez pratique car, d'une part, il permet de ne pas réécrire le système d'exploitation et donc de profiter de l'expérience des systèmes UNIX et, d'autre part, il minimise la mémoire utilisée par le noyau : les modules ne sont chargés que lorsque c'est nécessaire et ils sont déchargés quand le système n'en a plus besoin.

Le système d'exploitation Windows NT 4.0 est aussi un système mixte : son noyau (appelé Exécutif par Microsoft) contient un gestionnaire d'objet, un moniteur de références de sécurité, un gestionnaire de processus, une unité de gestion des appels de procédures locales, un gestionnaire de mémoire, un gestionnaire d'entrées / sorties et une autre partie que Microsoft nomme le noyau, probablement pour semer la confusion.

## Les services des systèmes d'exploitation

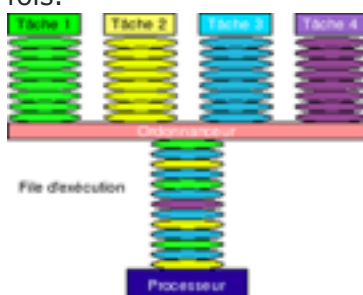
Notons les différents travaux que doit effectuer un système d'exploitation moderne. Ces travaux sont généralement nommés « services ». La plupart de ces travaux sont pris en charge par le noyau du système d'exploitation.

### La gestion des processus

La gestion des processus n'a de sens que sur les machines fonctionnant en temps partagé. Elle comprend la création et la destruction dynamique de processus. A noter qu'un processus est fondamentalement un programme qui s'exécute. Un processus se caractérise par un programme exécutable, ses données et sa pile d'exécution, son compteur ordinal, son pointeur de piles et autres registres. Un temps d'allocation processeur lui est alloué, quand celui-ci est dépassé le processus est alors suspendu. Comme il n'est arrêté que temporairement, il lui est assigné une adresse mémoire et une entrée dans la table des processus, ceci afin de le retrouver plus facilement pour le relancer.

### Gestion du processeur

Le processeur est l'unité centrale d'exécution d'un ordinateur. A chaque top de son horloge interne, soit plusieurs millions de fois par seconde, il exécute une instruction (ou une partie d'instruction). A un instant donné, un processeur ne peut donc effectuer qu'un traitement à la fois.



Pour obtenir un multitâche préemptif, le SE dispose d'un ordonnanceur qui gère l'ordre dans lequel les instructions des différents programmes sont exécutées. Des priorités peuvent être affectées à chaque tâche en fonction de leur importance.



## La gestion de la mémoire

Afin de simplifier la gestion des processus, les systèmes d'exploitation modernes travaillent dans un espace mémoire virtuel, c'est-à-dire avec des adresses virtuelles qui doivent être traduites pour correspondre à des adresses physiques.

Cela permet d'allouer à chaque processus (y compris au noyau) son propre espace mémoire de telle sorte qu'il a l'illusion d'avoir la machine pour lui tout seul. Cet espace est généralement découpé en trois segments : le segment correspondant au programme (*text* en anglais), le segment correspondant aux données (*data* en anglais) et le segment qui correspond à la mémoire utilisée en cours de route par le processus (généralement appelée la pile, *stack* en anglais).

Une autre partie de la gestion de la mémoire, très différente de celle présentée ci-dessus, concerne l'utilisation de pages de mémoire et la possibilité d'accéder à la mémoire secondaire pour optimiser la mémoire utilisée par tous les processus.

Donc elle gère en particulier le début/fin des travaux, ordre des travaux, les interruptions temporaires, le partage de temps de travail du processeur. Ainsi cette fonction assure la simultanéité des tâches dans les SE **multitâches**.

Pour fonctionner, les programmes occupent et utilisent de la place en mémoire. Le système d'exploitation gère l'allocation de cette mémoire aux applications. Quand la place en mémoire vive (RAM, Random Access Memory) vient à manquer, une partie du disque dur est utilisée comme mémoire virtuelle (plus lente que la RAM).

Un système d'exploitation consomme aussi de la RAM pour son propre usage. Les systèmes récents ont tendance à être de plus en plus "gourmands" en RAM. A titre d'exemple, Windows 7 nécessite 4 Go de RAM pour un usage confortable.

## La gestion des entrées / sorties

Les entrées / sorties permettent de faire transiter des données par l'ordinateur et d'utiliser ces données pour faire des calculs. Ces données peuvent provenir de périphériques, de processus présents sur la machine ou de processus présents sur d'autres machines (via un réseau). En fait cette fonction consiste à coordonner les liaisons entre la mémoire centrale et les différents périphériques via leurs interfaces. Elle gère l'enchaînement des ordres donnés aux périphériques, les interruptions (messages avec l'utilisateur) l'envoi et la réception de données. ...

## Le système de fichiers

Le système de fichiers est un élément essentiel des systèmes d'exploitation moderne : il permet d'accéder aux différents périphériques et il propose une interface abstraite pour manipuler des données.

Même si le système de fichiers fait souvent référence au disque dur, la notion de fichier est beaucoup plus générale.

Notamment bien comprendre son système d'arborescence de répertoires et de fichiers, avec un chemin (path) Racine (absolu) dès la première barre oblique (exemple C:/ ) et un chemin Relatif correspondant au fichier recherché (exemple C:/..../MonFichier.doc)

## La gestion des partitions

Un système d'exploitation doit piloter les différents périphériques de stockage présents sur l'ordinateur pour pouvoir y lire et écrire des données de manière persistante.

Un périphérique de stockage (disque dur, SSD, clé USB...) est divisé en zones spécifiques appelées partitions. Une partition est une unité de stockage logique, qui peut être formatée pour stocker des informations telles que des fichiers de données ou des applications. Pendant la phase d'installation, la plupart des systèmes d'exploitation partitionnent et formatent automatiquement l'espace disque.

# B SYSTEME DE GESTION DE FICHIERS : FAT 16, FAT 32, NTFS

## Stockage des fichiers

La gestion des données persistantes sur les périphériques de stockage est assurée par le système de gestion de fichiers (SGF) du système d'exploitation. Les fichiers sont découpés sous forme de blocs et répartis sur le support de stockage.

Un SGF offre les fonctionnalités suivantes :

- Manipulation des fichiers et des répertoires.
- Localisation des fichiers.
- Sécurité et contrôle des fichiers.

Il existe de nombreux SGF offrant des niveaux de fonctionnalités variables, comme FAT32 (File Allocation Table), NTFS (New Technology File System) ou ext4 (EXTended File System version 4). Le SGF est choisi lors du formatage d'une partition.

Le système d'exploitation propose également des mécanismes de gestion des fichiers et des répertoires : création, copie, suppression, formatage...

### 32 bits, 64 bits

Un système d'exploitation peut être disponible en versions 32 bits et 64 bits afin de s'adapter au processeur de l'ordinateur. Ce nombre de bits correspond à la taille des registres mémoire du processeur. L'augmentation de la largeur des registres permet, entre autres, d'adresser une quantité plus importante de mémoire vive. La quantité de RAM utilisable est limitée à 4 Go avec un processeur 32 bits. Un processeur 64 bits peut faire tourner un OS 32 bits, mais l'inverse n'est pas vrai.

Les systèmes d'exploitation de la famille Windows existent en version 64 bits depuis Windows XP.

### Propriétés de FAT16

- Supporte des partitions de tailles maximum 2.1 GO sous MS-DOS® et 4.0 GO sous Windows® NT 4.0
- Chaque partition peut contenir un maximum de 65,536 fichiers
- Limité à 512 entrées dans le répertoire principal du disque dur et 128 entrées dans le répertoire principal d'une disquette
- Pas de prise en charge des noms longs - La VFAT de Windows® 95/98 écrit des entrées supplémentaires dans une table d'allocation modifiée contenant les noms longs (maximum 255 caractères) ; ceci réduit le nombre maximum d'entrées dans le répertoire principal.
- **Avantages :**
  - Compatible avec les systèmes d'exploitation autres que DOS (dont Windows® 95, Windows® 98, et Windows® NT)
  - Pas de perte de place
- **Inconvénients :**
  - La grande taille des clusters réduit à une mauvaise gestion de l'espace disque pour les grandes partitions
  - Pas de compression disponible sous Windows® NT
  - Sécurité minimale - on peut uniquement jouer sur les attribut de fichiers (Lecture seule et Caché)
  - La mise à jour la table d'allocation des fichiers est lente - les performances diminuent avec l'augmentation de la taille des partitions si elle dépasse plusieurs centaines de MO

### Propriétés de FAT32

- Partition de taille maximum théorique de 2048 GO
- Chaque partition peut contenir un maximum de 2 097 152 fichiers
- Limité à 65,535 entrées dans le répertoire principal du disque dur
- Les noms de fichiers longs sont supportés sous VFAT Windows® 95/98 - VFAT écrit des entrées supplémentaires dans une table d'allocation modifiée contenant les noms longs (maximum 255 caractères) .
- **Avantages :**
  - Clusters de petite taille (jusqu'à 4K)
  - Répertoire principal ré-allocable permettant de stocker un grand nombre d'entrées
  - La **T**able d'**A**llocation des **F**ichiers (**FAT**) et le secteur de boot primaire (**M**aster **B**oot **R**ecord - **MBR**) peuvent être déplacés



- Des sauvegardes de la FAT et du MBR peuvent être effectuées avec les outils appropriés
- L'écriture sur la FAT secondaire peut être désactivée ; peut fonctionner directement à partir de la FAT secondaire si la principale n'a pas les bonnes informations concernant un secteur défectueux
- **Inconvénients :**
  - uniquement compatible avec Windows® 95 OSR2/98 - inaccessible si la partition FAT 32 est bootée depuis un média formaté sous un autre système d'exploitation autre que Microsoft Windows® 95 OSR2 ou Windows® 98
  - Petite perte de place - meilleure gestion des grands disques
  - Non compatible avec beaucoup d'utilitaires disques
- **Limitations :**
  - Sous Windows® 98, Windows® Me et Windows® 2000, la taille de la partition ne peut pas excéder 32 Go.

### Propriétés de NTFS

- Prise en charge de partition jusqu'à 16 exaoctets
- Niveau de sécurité C2 (homologué pour les usages gouvernementaux)
- Support des noms longs
- **Avantages :**
  - Clusters de petite taille
  - Les performances ne se dégradent pas avec l'augmentation de taille des partitions
  - Support de la compression de fichiers et de dossiers
  - Remappage automatique des mauvais clusters
  - Peu de maintenance disque pour les réparations
- **Inconvénients :**
  - Seulement compatible avec Windows® NT
  - Grande perte de place - ne doit pas être utilisé avec des disques inférieurs à 400 MO
  - Impossible de formater une disquette en système de fichiers NTFS

### Taille des clusters en fonction de la taille de la partition

	<b>FAT 16</b>	<b>FAT 32</b>	<b>NTFS</b>
<b>Taille des partitions</b>	<b>Taille Cluster</b>	<b>Taille Cluster</b>	<b>Taille Cluster</b>
16 MO - 128 MO	2K	***	512 octets
129 MO - 256 MO	4K	***	512 octets
257 MO - 512 MO	8K	***	512 octets
513 MO - 1024 MO	16K	4K	1K
1025 MO - 2048 MO	32K	4K	2K
2048 MO - 4096 MO	64K	4K	4K
4096 MO - 8192 MO	***	4K	4K
8192 MO - 16384 MO	***	8K	4K
16384 MO - 32768 MO	***	16K	4K
Supérieur à 32768 MO	***	32K	4K

**Nota Bene :** Les partitions FAT 16 de taille supérieure à 2.1 GO ne sont uniquement supportées que sous Windows® NT 4.0 . Pour Windows NT4.0 et 2000, la taille par défaut des clusters NTFS est de 4 KO.

## Propriétés/Caractéristiques 64 bits ... Faites vos recherches ...

/

/

/

/

/

/

### Conclusion : que retenir de ce chapitre ?

- un système d'exploitation est un programme ;
- un système d'exploitation est une machine virtuelle plus facile à programmer qu'une machine réelle ; à ce titre, il facilite l'accès aux ressources de la machine et il permet de développer des applications portables, qui ne sont pas spécifiques d'un ordinateur ou d'un système donné ;
- un système d'exploitation est un gestionnaire de ressources qui attribue les ressources aux différents utilisateurs et qui empêche l'accès illicite à celles-ci ; à ce titre, un système d'exploitation est garant du bon fonctionnement de la machine (fiabilité et stabilité) et de la conservation des données (pas de réamorçages intempestifs, pas de perdre de données) ;
- les appels systèmes sont l'interface que le système d'exploitation met à la disposition des utilisateurs pour qu'ils puissent lui demander des services ; sur les systèmes d'exploitation assurant leur rôle de gestionnaire de ressources, les utilisateurs ne peuvent pas accéder directement aux ressources de la machine et doivent nécessairement faire appel au système d'exploitation ;
- un système d'exploitation ne doit pas monopoliser le temps CPU de l'ordinateur qu'il gère et un bon système d'exploitation est capable d'effectuer son travail (notamment la gestion de l'accès aux ressources) sans avoir à disposer du processeur (utilisation des couches matériels comme la MMU) .
- **A continuer par vos soins ....**