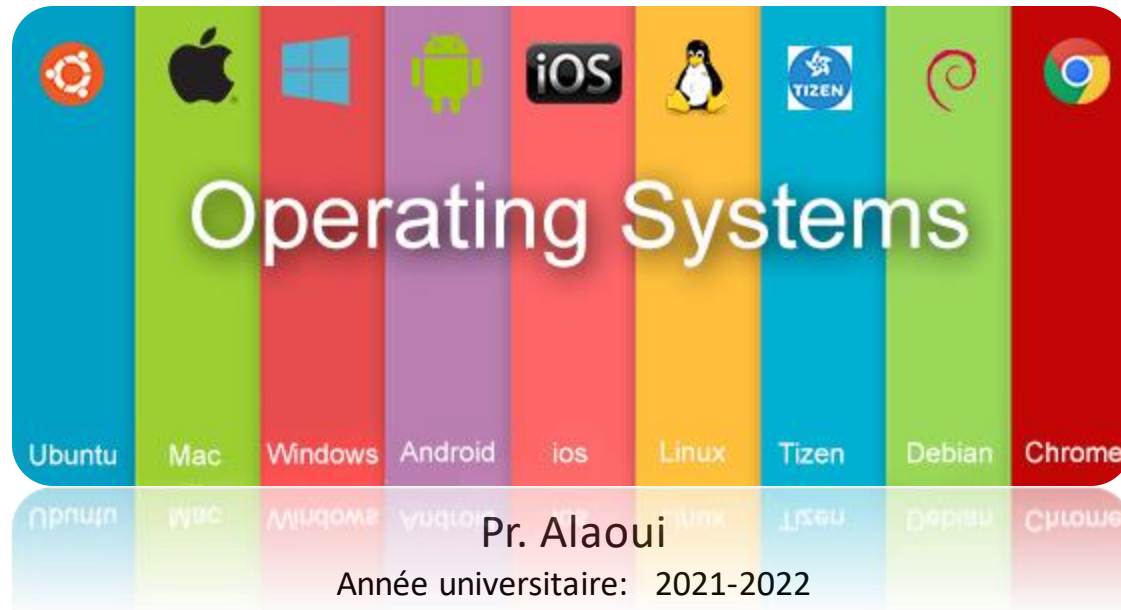


# M11

## Les Systèmes d'Exploitation



# Références bibliographiques

1. *Modern operating systems fourth edition*, Andrew s. Tanenbaum, Herbert bos.
2. *Cours et exercices corrigés, systèmes d'exploitation*, J. Archer Harris, ediscience.
3. *Conception de systèmes d'exploitation le cas linux*, Patrick Cegielski, deuxième édition.
4. *Systèmes d'exploitation des ordinateurs: principes de conception*, CROCUS, DUNOD.
5. *Operating system concepts*, Abraham Silberschatz and al., John wiley & sons.
6. *Cours de systèmes d'exploitation*, Pr. Hafid Bourzoufi, ISTV Valenciennes, France.
7. *Cours de systèmes d'exploitation*, Pr. Audrey Queudet, université de Nantes.
8. *Cours Operating Systems*, Pr. Geoffrey Challen, buffalo university, New York, USA.

# Motivations



Avant tout, quelles sont vos attentes de ce cours?

- Avez-vous déjà:
  - utilisé un système d'exploitation?
  - programmé sous un système d'exploitation donné?



# Motivations(2)

- **Au quotidien:**
  - Machines à laver, consoles de jeux, Smartphones, ordinateurs, SmartTV...
  - «Le système démarre» , «Le système s'est/a planté»  
#WindowsVista!, « version 3.0.1 du système », « bug système »,
- **Académique/Professionnel:**
  - Génie Logiciel: le logiciel s'appuie sur un OS (Operating System)
  - Administration réseau: Les machines du réseaux (PCs, routeurs, serveurs, ...) possèdent des OS!
- **Mais encore:** Les systèmes d'exploitation ont été particulièrement importants dans le développement de l'informatique, à côté de l'évolution technologique des ressources matérielles.

# Motivations(3)



- Ce cours permet d'accéder à des formations avancées (Systèmes distribués, virtualisation, cloud computing, Big Data ...).

Confrontation inévitable!

# Motivations(4)

- Avant d'aller plus loin:

*un ordinateur se compose d'éléments matériels « Hardware » et d'éléments logiciels « Software ».*

Cependant, comment coexistent-ils?

Quel est le rôle joué par le SE?

Comment réalise-t-il ce rôle?

Un premier contact:

- Un SE est intimement (fortement) lié aux ressources matérielles sur lesquelles il s'exécute.
- Il **étend l'ensemble d'instructions** exécutables par l'ordinateur (càd le matériel) et **gère ses ressources**.

# Plan général du cours

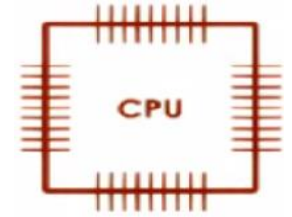
- I Généralités sur les SE
- II Gestion des processus
- III Communication interprocessus et synchronisation
- IV Gestion de la mémoire
- V Gestion des systèmes de fichiers

# I. Généralités sur les SE

1. Rappel: point sur le matériel
2. Rôles d'un SE
3. Définition d'un SE
4. Structure interne des SE
5. Le noyau
6. Quelques familles de SE
7. La notion de multitâches multi-utilisateurs



# Rappel: point sur le matériel!



## 1) Processeur:

Dit Unité Centrale de Traitement (UCT) ou processeur central (**CPU**), est « le cerveau » de l'ordinateur qui interprète et exécute les instructions du programme situées en mémoire centrale.

- L'UCT est composée de l'**Unité arithmétique et logique** (UAL) et de l'**Unité de commande** (ou de contrôle).
  - L'UAL effectue les opérations arithmétiques et logiques.
  - L'Unité de commande dirige le fonctionnement de toutes les autres unités: UAL, mémoire, entrées / sorties, etc., en leur fournissant les signaux de cadence (l'horloge) et de commande.

# Rappel: point sur le matériel!(2)

## 2)Mémoire:

Typical access time

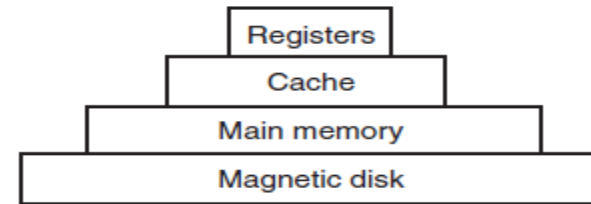
Typical capacity

1 nsec

2 nsec

10 nsec

10 msec



<1 KB

4 MB

1-8 GB

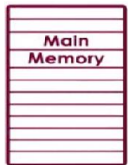
1-4 TB

- Les registres:

- type de mémoire qui sert à stocker des données à traiter, des résultats intermédiaires ou des informations de commande.
- Le temps d'accès est minimal, très petit espace de stockage et à prix élevé.

- La mémoire cache:

- Un bloc de cette mémoire est appelé **ligne de cache** qui est composée de plusieurs **mots mémoire** consécutifs en mémoire.
- Le processeur essaie d'accéder un mot d'abord dans la cache, avant de passer à la mémoire principale:
  - En cas d'échec (miss), le mot est gardé dans la cache pour un accès futur.
  - En cas de succès (hit), la mémoire principale n'est pas accédée.
- mémoire très rapide, de petite taille.



- La mémoire vive:

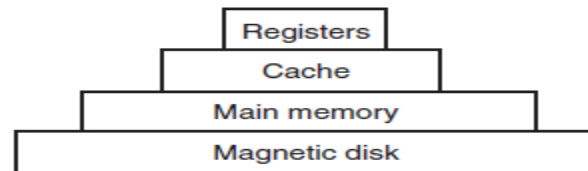
- RAM (Random Access Memory), on **peut accéder instantanément à n'importe quelle espace mémoire**.
- Volatile et contient les données et les instructions des applications en cours.
- Mémoire rapide, de taille plus importante et à prix moyen.

# Rappel: point sur le matériel!(2)

## 2)Mémoire:

Typical access time

1 nsec  
2 nsec  
10 nsec  
10 msec



Typical capacity

<1 KB  
4 MB  
1-8 GB  
1-4 TB

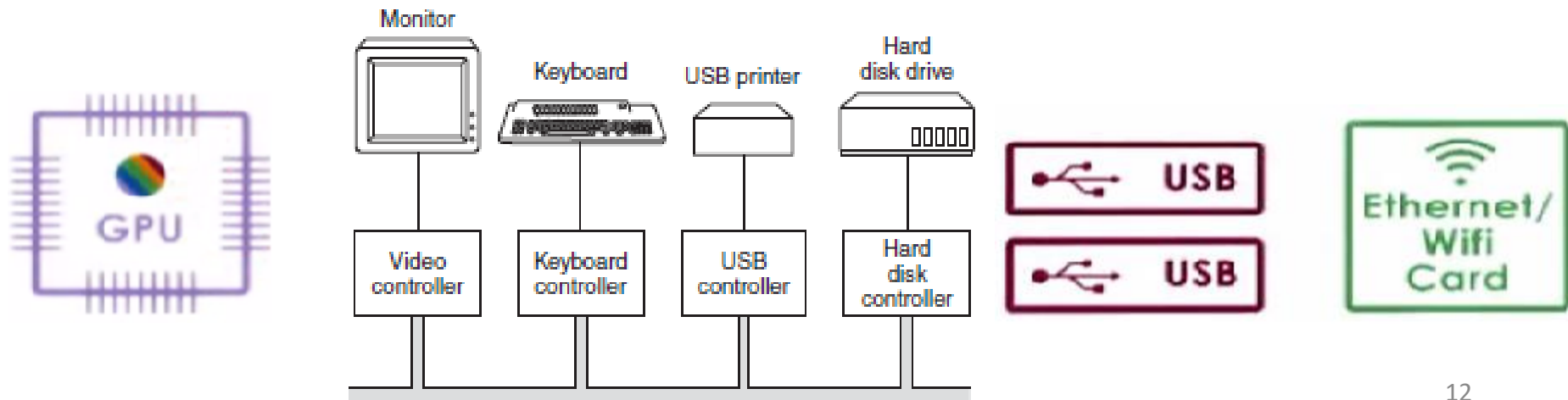


- **Le disque:**
  - sert principalement à stocker les données d'une manière non-volatile.
  - L'accès est mécanique, introduisant un délai important en lecture/écriture relativement à la RAM.
  - Sert éventuellement à « étendre » la RAM (à voir un peu plus loin).
  - Un espace plus important et à coût bas (selon les technologies).
- **La mémoire morte:**
  - ROM (Read Only Memory) **mémoire en lecture seule.**
  - Permanente contenant des microprogrammes enregistrés à l'usine sur des puces électroniques de la carte mère, contenant les routines de démarrage de l'ordinateur.
  - C'est une mémoire non volatile, rapide et à bas coût.

# Rappel: point sur le matériel!(3)

## 3) Les périphériques d'entrée/sortie (E/S):

- Permettent le dialogue (échange d'informations) avec ce qui se trouve à l'extérieur de la machine.
- Ils se composent généralement de deux parties:
  - **Le matériel** (physique).
  - **Le contrôleur**: une puce ou un ensemble de puces qui contrôle physiquement le périphérique. Présente au système d'exploitation une interface « simple », nommée Driver, pour recevoir les commandes et retourner leurs résultats.



# Le système d'exploitation est un gérant!



**Gérant**

- **Dirige ses ressources**
  - Répartition des tâches, des commandes, des outils de travail, temps de travail, ...
- **Fait respecter la politique de travail**
  - La coopération entre collègues, processus métiers, ...
- **Atténue la difficulté des tâches complexes**
  - Simplifie les opérations, optimise les performances, ...

# Le système d'exploitation est un gérant!(2)

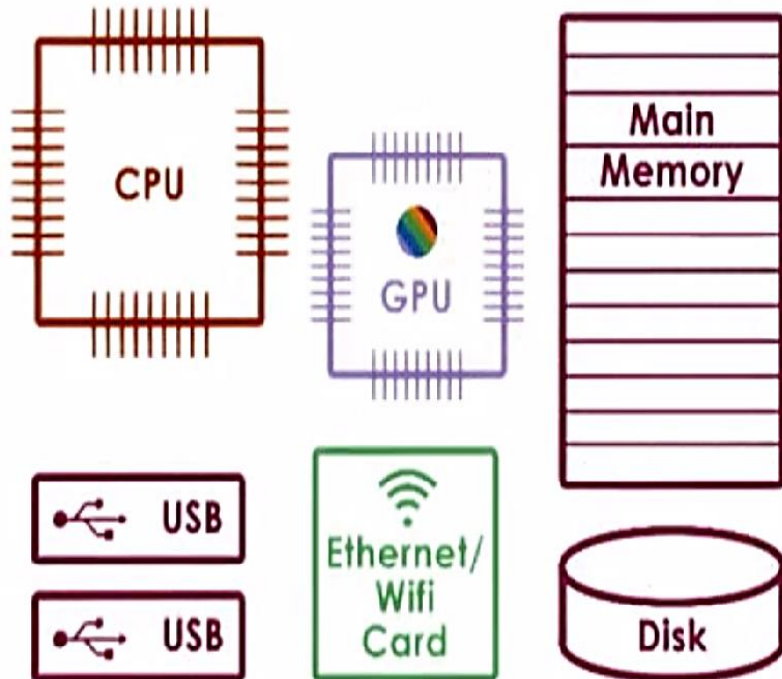


**Système d'exploitation**

- **Dirige ses ressources**
  - Utilisation/allocation de la CPU, mémoire, périphériques, ...
- **Fait respecter la politique de travail**
  - Priorité entre processus, limiter la monopolisation, ouverture de fichiers, ...
- **Atténue la difficulté des tâches complexes**
  - Simplifier l'interaction avec le matériel, ...

# Les rôles d'un SE

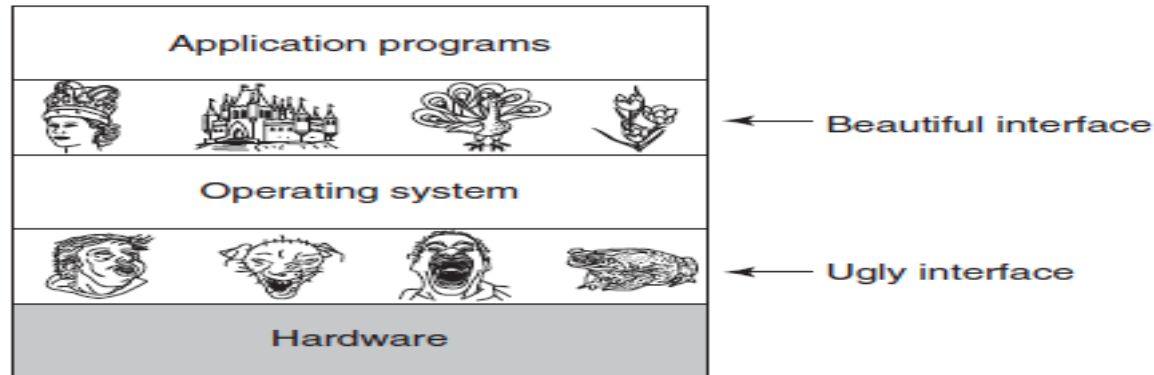
« Il est plus facile de définir un système d'exploitation par ce qu'il fait que par ce qu'il est. »



- Cache la complexité du hardware, que ce soit pour les applications ou pour les développeurs: diverses technologies de stockage, de transmission ... .
- Gère les ressources: allocation de mémoire, ordre d'exécution.
- Isolation et protection: l'exécution des programmes, les utilisateurs, ...

# Les rôles d'un SE(2)

- D'une manière simple: Faire abstraction (simplifier l'interaction) du matériel puis contrôler et gérer son utilisation.



- D'une manière plus soutenue (académique):
  - **La gestion des processus**: un processus est un programme en cours d'exécution. Le SE doit gérer l'allocation de ressources aux processus en proposant à chacun un environnement dans lequel il peut s'exécuter en toute sécurité.

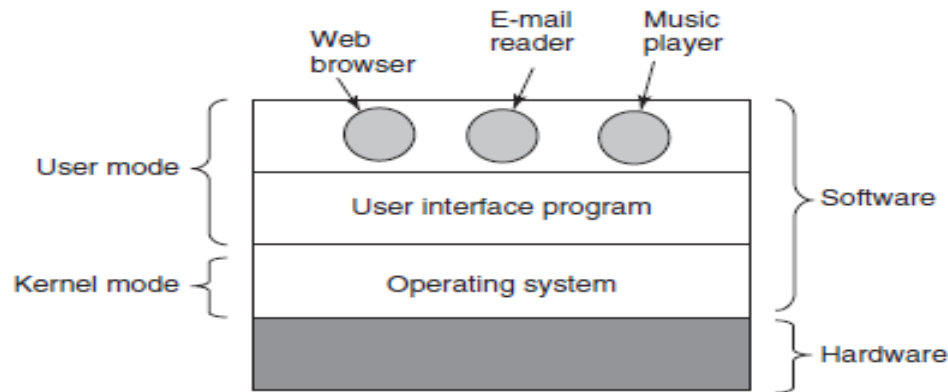


# Les rôles d'un SE(3)

- **La gestion de la mémoire** : Le SE doit gérer l'allocation de mémoire aux processus et contrôler physiquement les emplacements auxquels peut accéder un processus.
- **La gestion des périphériques**: Le SE doit gérer les périphériques afin de leur permettre d'être partagés de manière efficace entre les processus.
- **La gestion du système de fichiers** : Le SE doit gérer des structures de données permettant de créer, stocker, supprimer, lire (etc. ...) les informations et de les organiser dans des fichiers sur des mémoires secondaires (disque dur, disquette, CD-ROM, clé USB, disques SSD, etc.).
- **La protection**: l'accès aux données doit être réglementé puisqu'il existe plusieurs utilisateurs et processus. Le SE doit garantir que les fichiers, les segments de mémoire, etc ... ne peuvent être utilisés que par les processus ayant obtenu l'autorisation appropriée.
- **La sécurité**: Le SE doit posséder des mécanismes pour se défendre contre les attaques externes et internes. Elles comprennent les virus, des attaques de déni de service (Denial of Service: DoS).

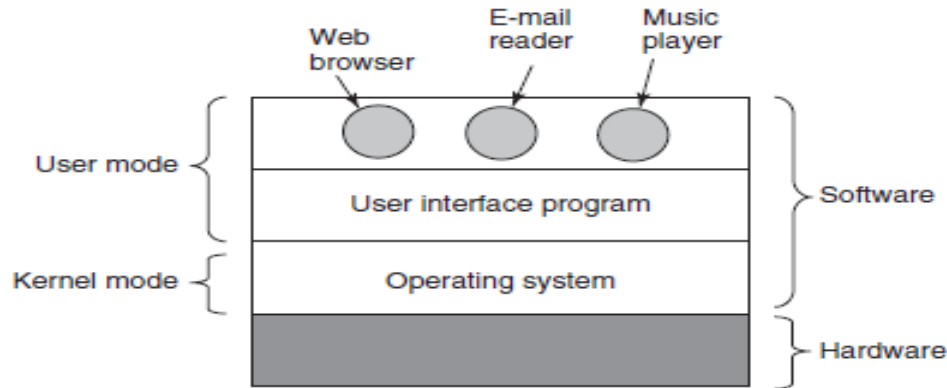
# Définition d'un SE

- **Définition:** *Un système d'exploitation est une couche logicielle indispensable pour exploiter, d'une manière simple, les ressources matériels d'un ordinateur.*



- Deux modes de fonctionnement (en général):
  - le mode noyau (mode superviseur, privilégié): il dispose d'un accès complet à tout le matériel et peut exécuter toutes les instructions que la machine est capable d'exécuter.
  - Le mode utilisateur: le reste du système s'exécute en mode utilisateur, dans lequel seul un sous-ensemble des instructions de la machine est exécutable.

# Définition d'un SE(3)



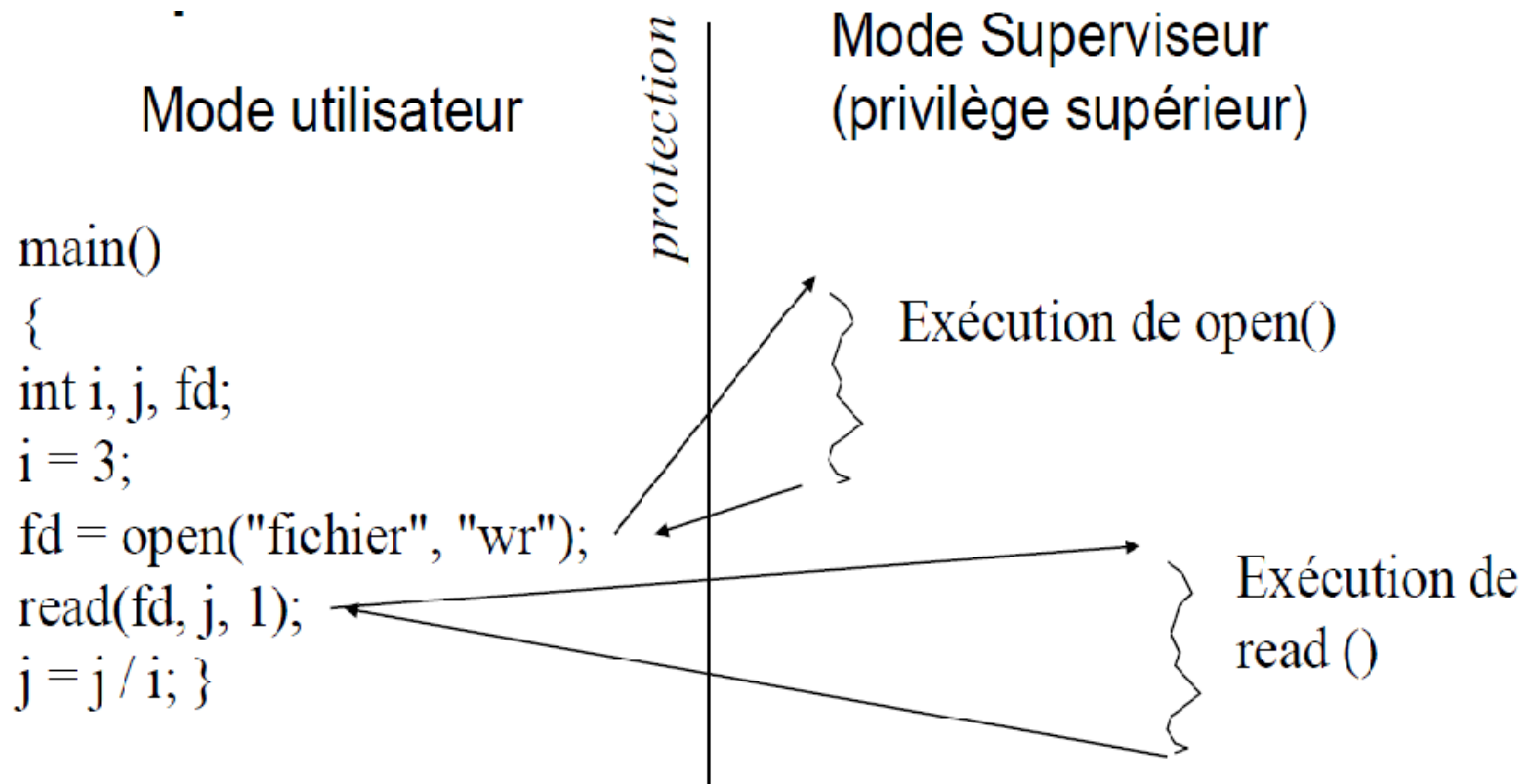
- **Le programme d'interface utilisateur:** le shell ou l'interface graphique, est le niveau le plus bas du logiciel en mode utilisateur, et permet à l'utilisateur de démarrer d'autres programmes, comme un navigateur Web, un lecteur de courrier électronique ou un lecteur de musique...

# Structures internes de SE

- **Monolithique (d'un seul bloc) :**
  - L'ensemble du système d'exploitation s'exécute en un seul programme en mode noyau.
  - Le système d'exploitation est écrit comme une collection de procédures, reliées entre elles dans un seul programme binaire exécutable unique.
  - ✓ **Implémentation simple.**
  - ⊗ **Difficile à maintenir!**
  - MS-DOS est un exemple d'un tel système.
- **Systèmes à modes noyau et utilisateur:**
  - Le système d'exploitation démarre en mode noyau, ce qui permet d'initialiser les périphériques et de mettre en place des routines de service, et commute ensuite en mode utilisateur.
  - En mode utilisateur, on utilise les **appels système** pour avoir accès à ce qui a été prévu par le système.
  - Unix et Windows (tout au moins depuis Windows 95) sont de tels systèmes.

# Structures internes des SE(2)

- Systèmes à modes noyau et utilisateur: Appel système



- 1) Le noyau reçoit l'appel système,
- 2) Vérifie qu'il s'agit d'une demande valable (en particulier du point de vue des droits d'accès),
- 3) Exécute,
- 4) Renvoie au mode utilisateur.

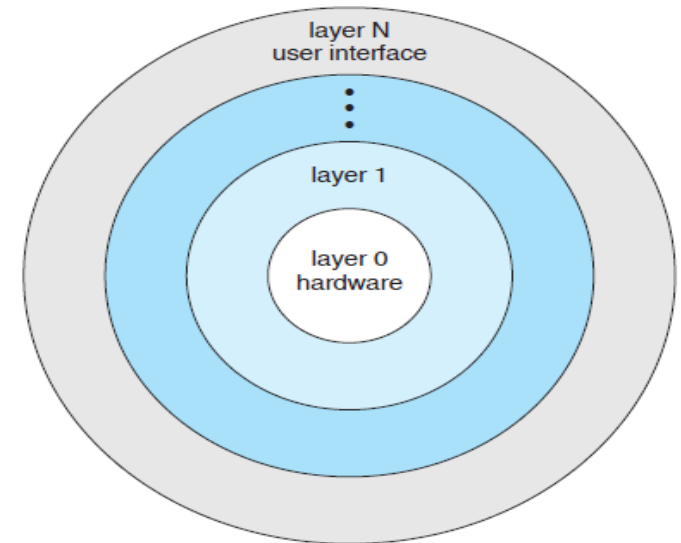
# Structures internes des SE(3)

## Systèmes en couches (généralisation):

- Chaque couche réalise sa fonction en s'appuyant exclusivement sur l'ensemble des fonctions de la couche qui lui est immédiatement inférieure.
- Une couche typique du système d'exploitation consiste en des structures de données et un ensemble de routines pouvant être invoquées par des couches de niveau supérieur.

✓ Vérification et débogage relativement simples.

⊗ Difficulté lors de la définition des couches; tendance à être moins efficaces.

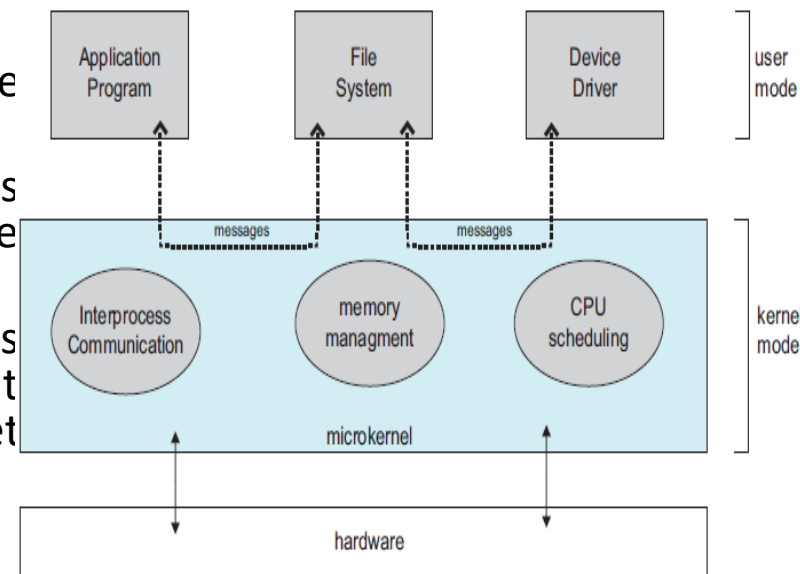


## • Systèmes à micro Kernel:

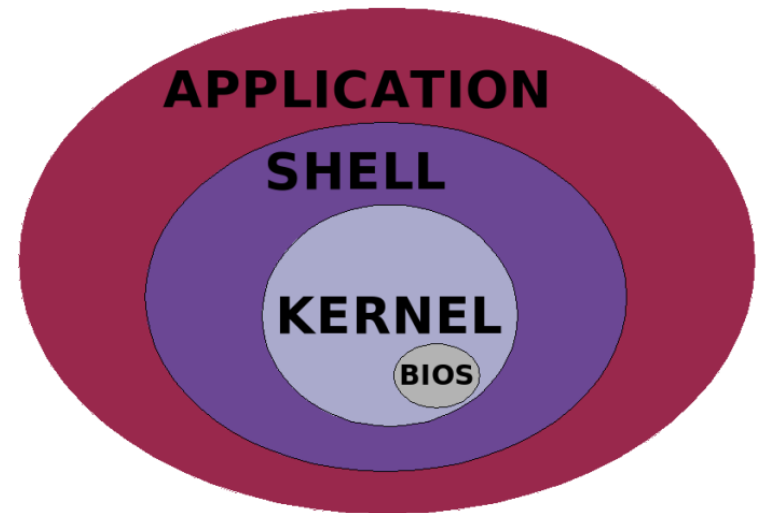
- Un noyau minimal (appelé micronoyau) de taille de code réduite.
- Se contente en général de gestionnaires de tâches et de mémoire simples, et un mécanisme de communication entre processus.
- Les gestionnaires de périphériques, les gestionnaires d'appels système, etc... sont implémentés en tant que programmes système et utilisateurs.

✓ Un SE plus simple à étendre.

⊗ Problèmes de performance.



# Le noyau



- Le noyau (kernel en anglais) comporte un certain nombre des plus importantes routines (sous-programmes) du SE. Il est chargé en mémoire vive à l'initialisation du système. Les autres routines, moins critiques, sont appelées des **utilitaires**.
- Le noyau d'un système d'exploitation se compose de quatre parties principales : **le gestionnaire de tâches** (ou des processus), **le gestionnaire de mémoire**, **le gestionnaire de fichiers** et **le gestionnaire de périphériques d'entrée-sortie**.
- Il possède également deux parties auxiliaires : **le chargeur du système d'exploitation** et **l'interpréteur de commandes**.

# Le noyau(2)

- **Gestionnaire de tâches (ordonnanceur) :**
  - Divise le temps en laps de temps (en anglais **slices**, tranches),
  - Décide périodiquement d'interrompre le processus en cours et de démarrer (ou reprendre) l'exécution d'un autre.
- **Gestionnaire de mémoire:**
  - Connaître les parties libres et les parties occupées de la mémoire,
  - Allouer de la mémoire aux processus qui en ont besoin,
  - Récupérer la mémoire utilisée par un processus lorsque celui-ci se termine,
  - Traiter le va-et-vient entre le disque et la mémoire principale lorsque cette dernière doit être étendue.
- **Gestionnaire de fichiers:**
  - Faire abstraction des spécificités des disques et des autres périphériques d'entrée-sortie,
  - Offrir au programmeur un modèle agréable et facile d'emploi.



# Le noyau(3)

- Gestionnaire de périphériques:
  - Envoyer les commandes aux périphériques,
  - Intercepter les interruptions,
  - Traiter les erreurs.
- Le chargeur du système d'exploitation:
  - Appelé, pour PC et MAC, BIOS (pour Basic Input Output System) et est chargé à une adresse bien déterminée en mémoire RAM.
  - Ce logiciel initialise les périphériques, charge un secteur du disque, et exécute ce qui y est placé.
- L'interpréteur de commandes (shell en anglais):
  - Est souvent considéré comme une partie du SE
  - Exécute une boucle infinie qui affiche une invite (montrant par là que l'on attend quelque chose), lit le nom du programme et les paramètres saisis par l'utilisateur à ce moment-là et l'exécute.

# Quelques familles de SE

- L'existence des SE depuis fort longtemps ainsi que leur utilisation dans différents domaines technologiques ont permis l'émergence de plusieurs familles de SE.
- **Mainframe\*:**
  - Ordinateurs de taille d'une pièce qu'on trouve dans les centres de données d'entreprise.
  - Grandes capacités: 1000 disques et des millions de gigaoctets de données.
  - Les SE sont amenés à traiter plusieurs tâches à la fois, dont la plupart nécessitent des quantités importantes d'E/S.



# Quelques familles de SE

- Serveurs:

- PC à ressources importantes ou des workstations ou des Mainframes.
- Les SE doivent être capables de gérer plusieurs utilisateurs à la fois sur un réseau et de partager des ressources matérielles et logicielles.



- Multiprocesseurs:

- Un moyen pour obtenir une puissance de calcul majeure est de connecter plusieurs CPU dans un seul système.
- Ces systèmes ont besoin de SE assez spéciaux, des variations de SE pour serveurs.

# Quelques familles de SE(2)

- **Personal Computer:** Les SE pour PC modernes supportent la multiprogrammation, pour fournir un bon support à un seul utilisateur.

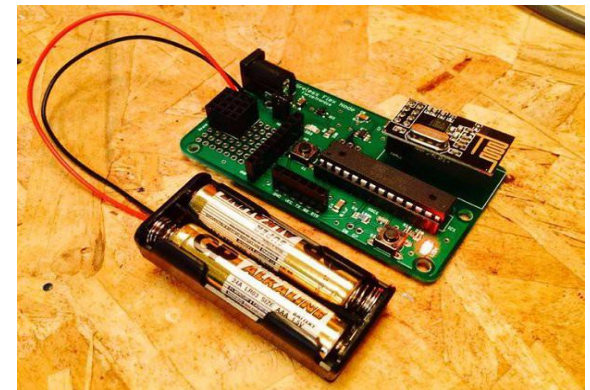
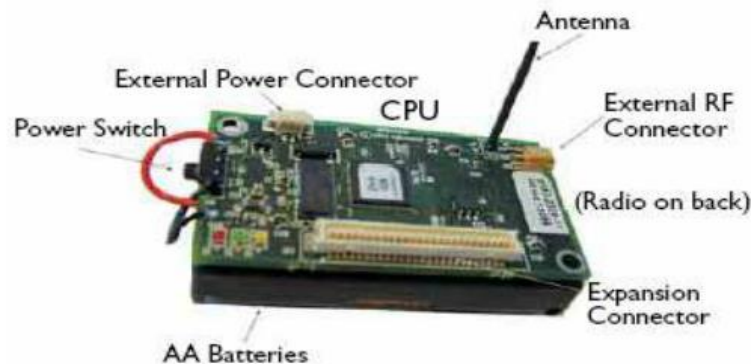


- **Handheld:**
  - Connu auparavant sous le nom de PDA (Personal Digital Assistant), ancêtre des smartphones et des tablettes.
  - La plupart de ces appareils possèdent des CPU multicoeurs, GPS, caméras, capteurs...
  - Ils ont des exigences particulières ce qui nécessite des SE sophistiqués et bien adaptés.



# Quelques familles de SE(3)

- Les systèmes embarqués:
  - S'exécutent sur des circuits pour des périphériques n'acceptant pas de futures installations de logiciels par l'utilisateur (Ex: fours à micro-ondes, les téléviseurs, les voitures, etc ): tout est en ROM.
  - Ces SE n'ont pas besoin de considérer l'aspect de protection contre les logiciels malveillants, ce qui entraîne une simplification de la conception.
- Nœud capteur\*:
  - Unités qui composent un réseau de capteurs sans fil.
  - Chaque nœud est composé de CPU, RAM, ROM, capteurs (température, humidité, lumière, mouvement ... ): un vrai ordinateur.
  - Le SE doit être petit et simple car les nœuds ont peu de RAM et sont contraints en énergie.



# Quelques familles de SE(4)

- **Systèmes temps réel\***:
  - Se caractérisent par le temps (ou le délai) comme contrainte.
  - Si le système doit fournir des garanties absolues qu'une certaine action se produira à un certain moment (ou dans un certain intervalle temps), on parle d'un système en temps réel strict (**Hard real-time**).
  - Si le système doit fournir des garanties avec une certaine probabilité et ainsi un délai plus grand mais acceptable, on parle de temps réel souple (**Soft real-time**).



# La notion de multitâches multi-utilisateurs

- **Systèmes multitâches:**

- Appelé aussi multi-programmé
- Permet l'exécution de plusieurs tâches à la fois : exécuter un programme utilisateur, lire les données d'un disque, afficher des résultats sur un terminal.
- Ce contexte fait appel aux notions suivantes:
  - **Processus (et non programme):** un processus est une instance de programme en exécution. Représenté par un programme (le code), ses données et son état d'avancement communément appelés variables d'environnement.
  - **Temps partagé:** le micro-processeur à un instant donné, n'exécute réellement qu'un seul processus. Faire passer le processeur d'un processus à un autre, en exécutant chaque programme pendant quelques dizaines de millisecondes, donne l'impression que tout s'exécute en même temps : **Le pseudo-parallélisme.**
  - **Espace mémoire d'un processus:** chaque processus possède son propre espace mémoire, non accessible aux autres processus. On parle de **l'espace d'adressage du processus.**

# La notion de multitâches multi-utilisateurs(2)

- **Systèmes multi-utilisateurs:**
  - Capable d'exécuter de façon **concurrente** et **indépendante** des applications appartenant à plusieurs utilisateurs.
  - **Concurrente?:**
    - les applications sont actives au même moment et se disputent l'accès aux différentes ressources.
  - **Indépendante?:**
    - chaque application peut réaliser son travail sans se préoccuper de ce que font les applications des autres utilisateurs.
  - Ce contexte fait appel aux notions suivantes:
    - **Utilisateurs:** matérialisé par un espace privé de travail sur la machine. Chaque utilisateur est identifié par un numéro unique, appelé l'identifiant de l'utilisateur, ou UID (pour l'anglais User Identifier).
    - **Groupe d'utilisateurs:** permet de partager de façon sélective le matériel avec d'autres utilisateurs. Un groupe est également identifié par un numéro unique dénommé identifiant de groupe, ou GID (pour l'anglais Group Identifier).
    - **Super-utilisateur:** ou encore superviseur (root en anglais), un utilisateur particulier qui peut faire pratiquement tout dans la mesure où le système d'exploitation ne lui applique jamais les mécanismes de protection. Il peut, en particulier, accéder à tous les fichiers du système et interférer sur l'activité de n'importe quel processus en cours d'exécution.