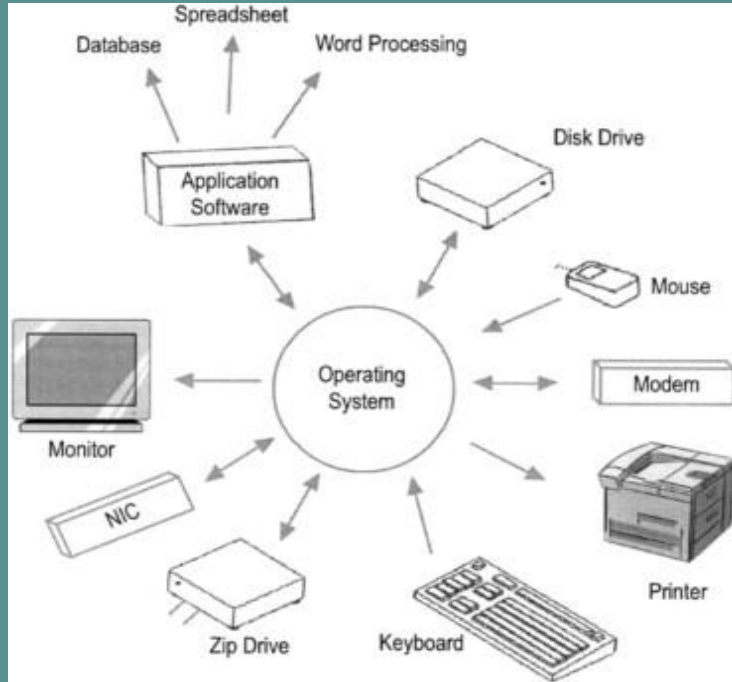
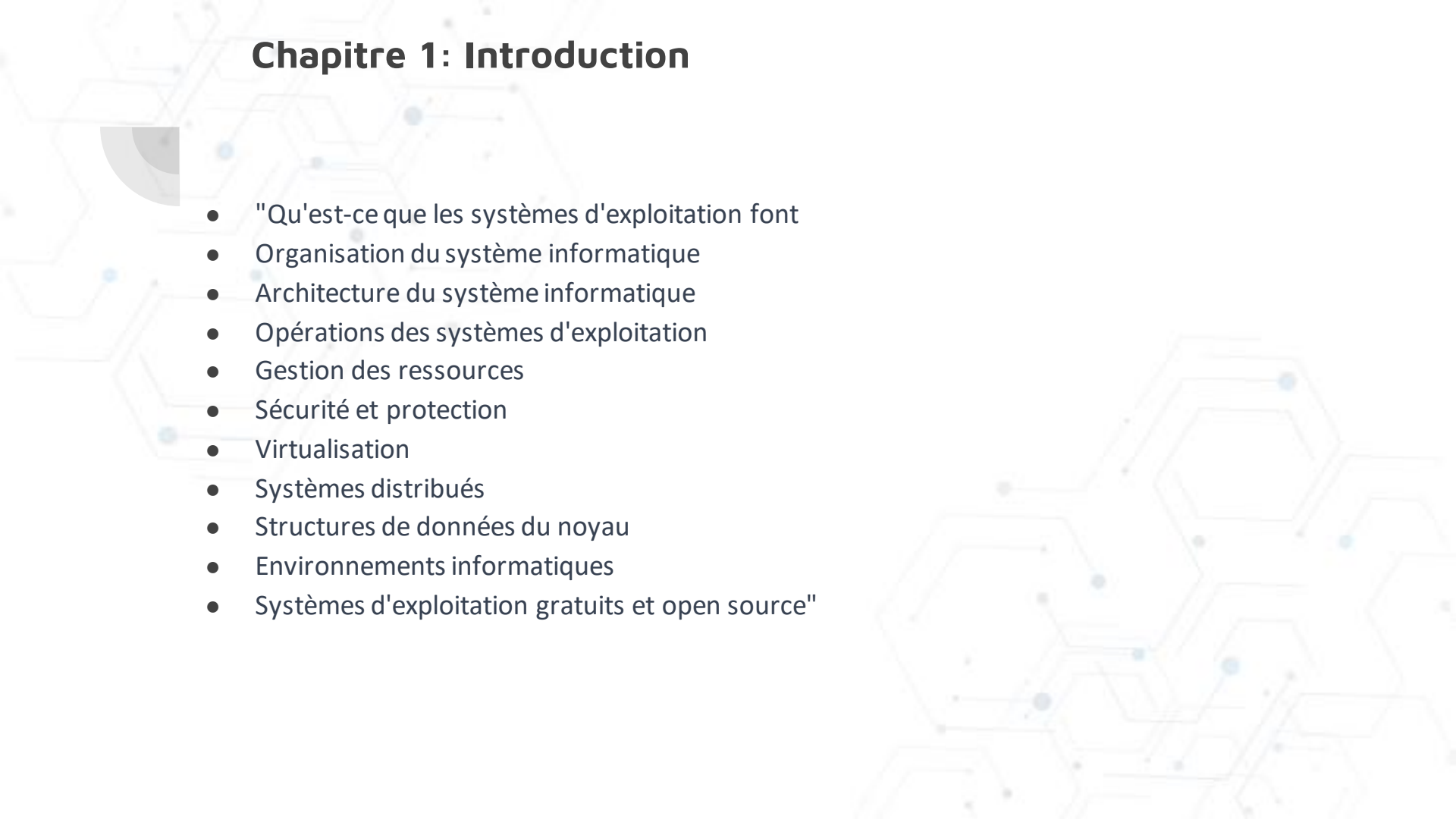


# Operating Systems



Prof. B. A. Bagula  
Department of Computer Science,  
University of the Western Cape, South Africa.

# Chapitre 1: Introduction

- 
- "Qu'est-ce que les systèmes d'exploitation font
  - Organisation du système informatique
  - Architecture du système informatique
  - Opérations des systèmes d'exploitation
  - Gestion des ressources
  - Sécurité et protection
  - Virtualisation
  - Systèmes distribués
  - Structures de données du noyau
  - Environnements informatiques
  - Systèmes d'exploitation gratuits et open source"

# Objectives


- "Décrire l'organisation générale d'un système informatique et le rôle des interruptions.
- Décrire les composants d'un système informatique moderne multiprocesseur.
- Illustrer la transition du mode utilisateur au mode noyau.
- Discuter de la manière dont les systèmes d'exploitation sont utilisés dans divers environnements informatiques.
- Fournir des exemples de systèmes d'exploitation gratuits et open source."

# What Does the Term Operating System Mean?



- Un système d'exploitation est à un ordinateur ce qu'un « complétez les blancs » est à d'autres appareils :
- Voiture :
- Avion :
- Imprimante :
- Machine à laver :
- Grille-pain :
- Compilateur :

# What Does the Term Operating System Mean?



Un système d'exploitation est à un ordinateur ce qu'un « complétez les blancs » est à d'autres appareils :

- Voiture : Un système d'exploitation est à un ordinateur ce qu'une unité de commande moteur (ECU) est à une voiture.
- Avion : Un système d'exploitation est à un ordinateur ce qu'un système de contrôle de vol est à un avion.
- Imprimante : Un système d'exploitation est à un ordinateur ce qu'un pilote d'imprimante est à une imprimante.
- Machine à laver : Un système d'exploitation est à un ordinateur ce qu'une unité de contrôle est à une machine à laver.
- Grille-pain : Un système d'exploitation est à un ordinateur ce qu'un minuteur et un contrôle d'élément chauffant sont à un grille-pain.
- Compilateur : Un système d'exploitation est à un ordinateur ce qu'un interpréteur de langage est à un compilateur.

Dans chaque analogie, le système d'exploitation joue un rôle crucial dans la gestion et la coordination des différents composants ou fonctions de l'appareil, de manière similaire à la façon dont d'autres systèmes ou composants remplissent des fonctions spécifiques dans différents appareils.

# What is an Operating System?

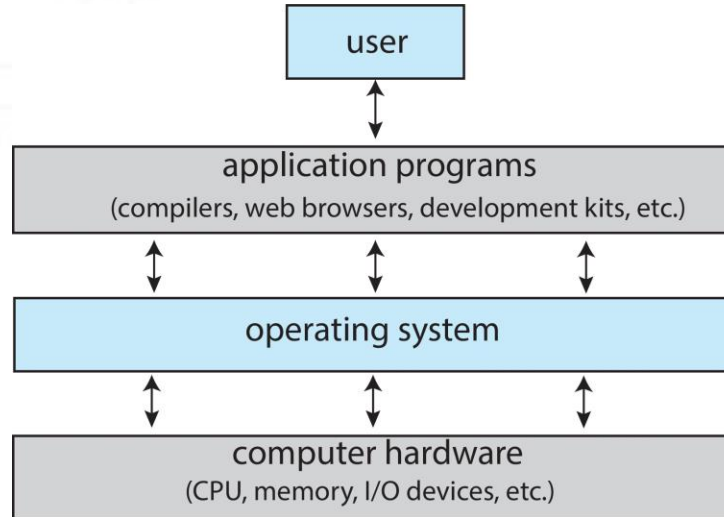


- Un programme qui agit comme un intermédiaire entre un utilisateur d'un ordinateur et le matériel informatique de l'ordinateur est appelé un "système d'exploitation".
- Objectifs du système d'exploitation :
  - Exécuter des programmes utilisateur et faciliter la résolution des problèmes de l'utilisateur.
  - Rendre le système informatique pratique à utiliser.
  - Utiliser de manière efficace le matériel informatique de l'ordinateur.

# Computer System Structure

- Le système informatique peut être divisé en quatre composants principaux :
  1. Matériel - fournit des ressources informatiques de base
    - Processeur (CPU), mémoire, périphériques d'entrée/sortie (I/O)
  2. Système d'exploitation
    - Contrôle et coordonne l'utilisation du matériel entre différentes applications et utilisateurs
  3. Programmes d'application
    - Définissent la manière dont les ressources système sont utilisées pour résoudre les problèmes informatiques des utilisateurs
    - Exemples : traitement de texte, compilateurs, navigateurs web, systèmes de bases de données, jeux vidéo
  4. Utilisateurs
    - Personnes, machines, autres ordinateurs

# Abstract View of Components of Computer






# What Operating Systems Do

- Cela dépend du point de vue.
- Les utilisateurs veulent la commodité, la facilité d'utilisation et de bonnes performances.
  - Ils ne se préoccupent pas de l'utilisation des ressources.
- Mais un ordinateur partagé tel qu'un ordinateur central ou un mini-ordinateur doit satisfaire tous les utilisateurs.
  - Le système d'exploitation est un allocateur de ressources et un programme de contrôle qui utilise efficacement le matériel et gère l'exécution des programmes utilisateur.
- Les utilisateurs de systèmes dédiés tels que des stations de travail ont des ressources dédiées mais utilisent fréquemment des ressources partagées à partir de serveurs.
- Les appareils mobiles tels que les smartphones et les tablettes ont peu de ressources, optimisées pour la convivialité et la durée de vie de la batterie.
  - Les interfaces utilisateur mobiles incluent des écrans tactiles, la reconnaissance vocale.
- Certains ordinateurs ont peu ou pas d'interface utilisateur, tels que les ordinateurs intégrés dans des appareils et des automobiles.
  - Ils fonctionnent principalement sans intervention de l'utilisateur.

# Defining Operating Systems

- Le terme "système d'exploitation" englobe de nombreux rôles
  - en raison de la myriade de conceptions et d'utilisations des systèmes d'exploitation.
  - Présent des grille-pains aux navires, aux engins spatiaux, aux consoles de jeux, aux télévisions et aux systèmes de contrôle industriel,
  - il est apparu lorsque les ordinateurs à usage militaire, initialement dédiés, sont devenus plus généraux et ont nécessité une gestion des ressources et un contrôle des programmes.

# Operating System Definition

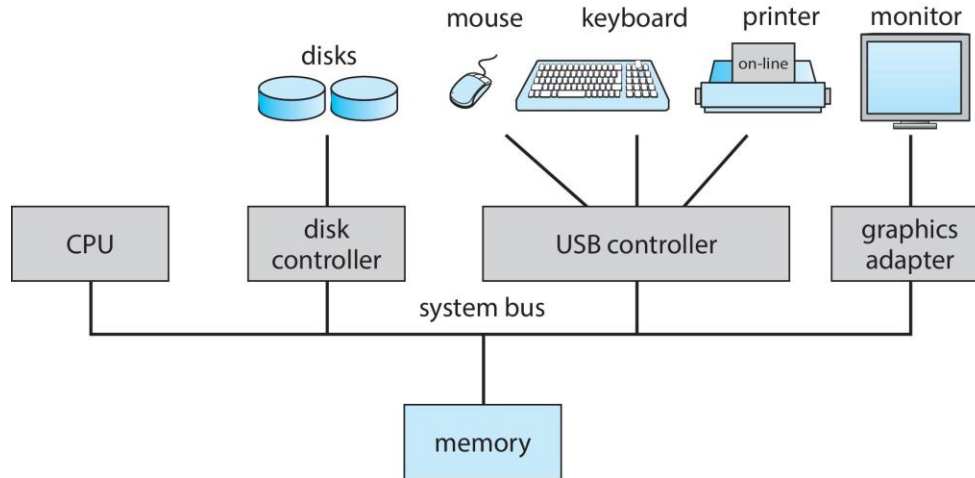
- 
- Il n'existe pas de définition universellement acceptée.
  - " Tout ce qu'un fournisseur expédie lorsque vous commandez un système d'exploitation " est une bonne approximation,
    - mais cela varie considérablement. "
  - Le programme qui s'exécute en permanence sur l'ordinateur " est le noyau, faisant partie du système d'exploitation.
  - Tout le reste est soit
    - un programme système (livré avec le système d'exploitation, mais non partie du noyau), soit
    - un programme d'application, c'est-à-dire tous les programmes non associés au système d'exploitation.
  - Les systèmes d'exploitation d'aujourd'hui, destinés à un usage général et à l'informatique mobile, incluent également des intergiciels - un ensemble de cadres logiciels qui fournissent des services supplémentaires aux développeurs d'applications, tels que les bases de données, le multimédia et les graphiques.



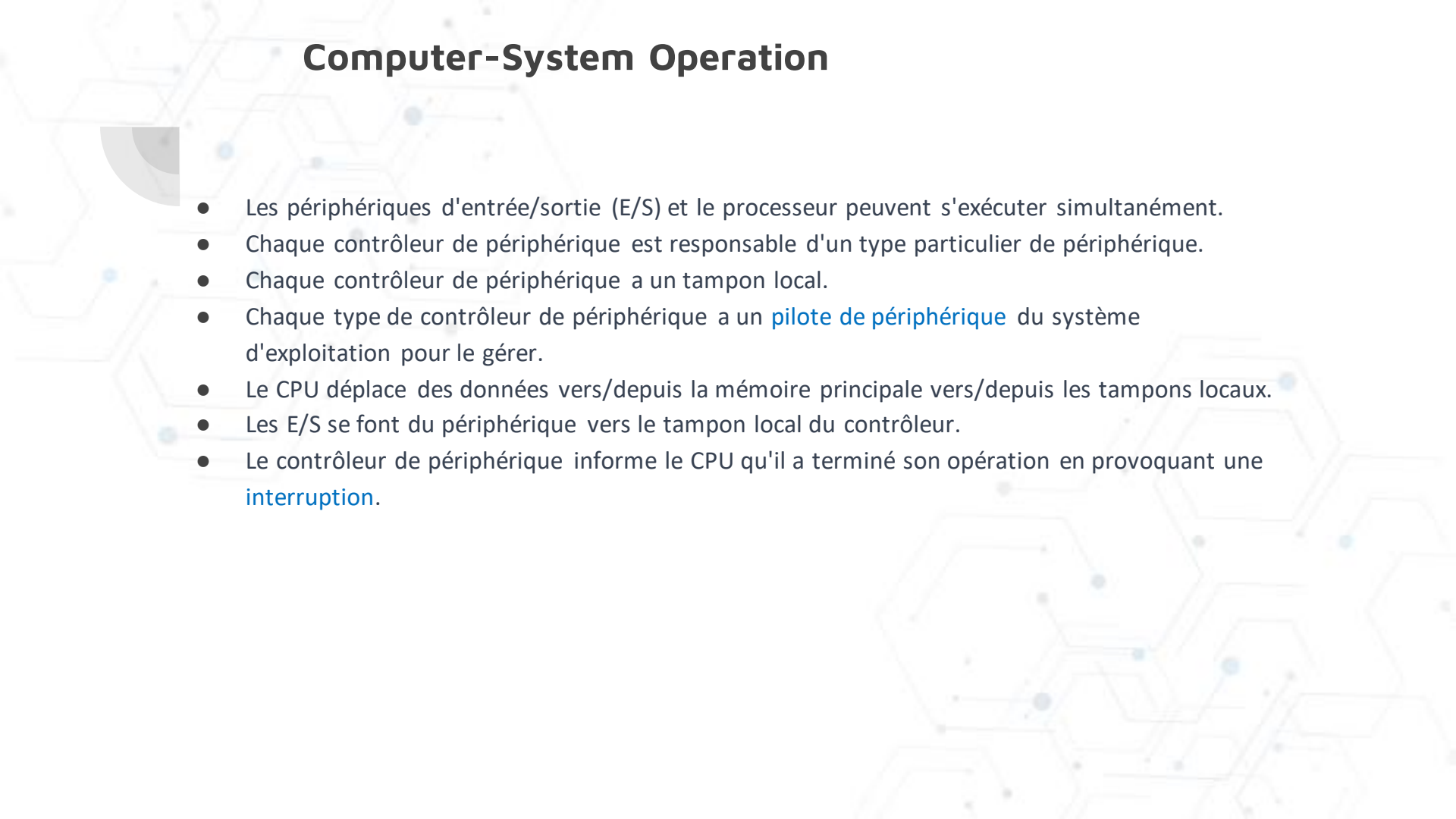

# Overview of Computer System Structure

# Computer System Organization

- Fonctionnement du système informatique
  - Un ou plusieurs processeurs (CPU), des contrôleurs de périphériques connectés par le biais d'un **bus** commun fournissant un accès à une mémoire partagée.
  - Exécution simultanée des CPUs et des dispositifs concurrents pour les cycles mémoire.



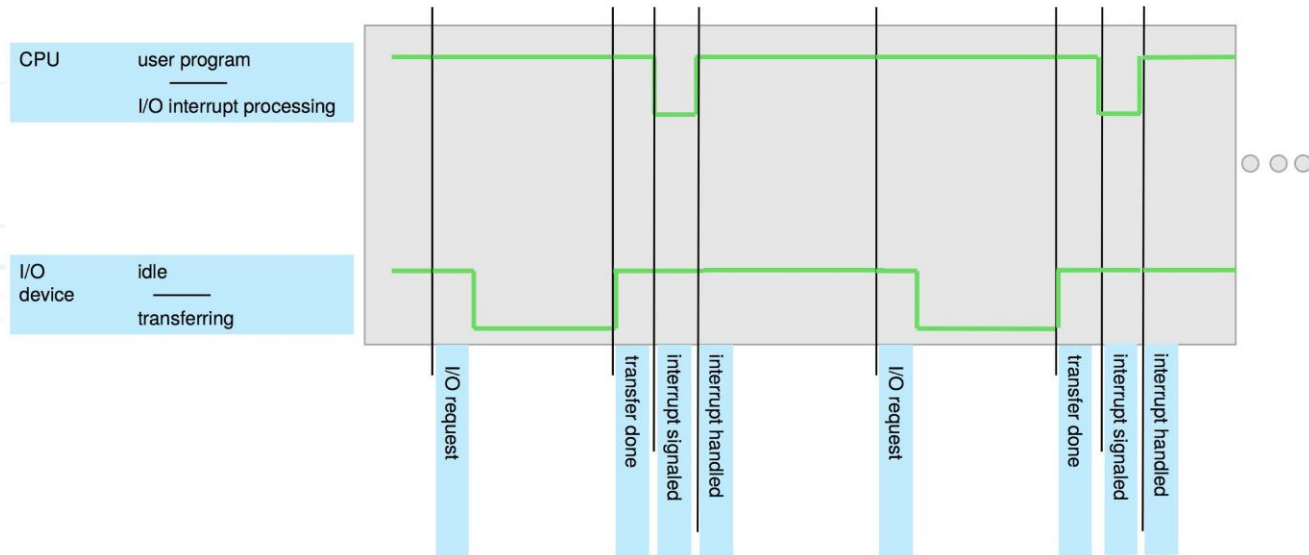
# Computer-System Operation

- 
- 
- Les périphériques d'entrée/sortie (E/S) et le processeur peuvent s'exécuter simultanément.
  - Chaque contrôleur de périphérique est responsable d'un type particulier de périphérique.
  - Chaque contrôleur de périphérique a un tampon local.
  - Chaque type de contrôleur de périphérique a un **pilote de périphérique** du système d'exploitation pour le gérer.
  - Le CPU déplace des données vers/depuis la mémoire principale vers/depuis les tampons locaux.
  - Les E/S se font du périphérique vers le tampon local du contrôleur.
  - Le contrôleur de périphérique informe le CPU qu'il a terminé son opération en provoquant une **interruption**.

# Common Functions of Interrupts

- L'interruption transfère le contrôle vers la routine de service d'interruption généralement, à travers le **vecteur d'interruption**, qui contient les adresses de toutes les routines de service.
- L'architecture d'interruption doit sauvegarder l'adresse de l'instruction interrompue.
- Une **exception** est une interruption générée par le logiciel, causée soit par une erreur, soit par une demande de l'utilisateur.
- Un système d'exploitation **fonctionne grâce aux interruptions**.

# Interrupt Timeline



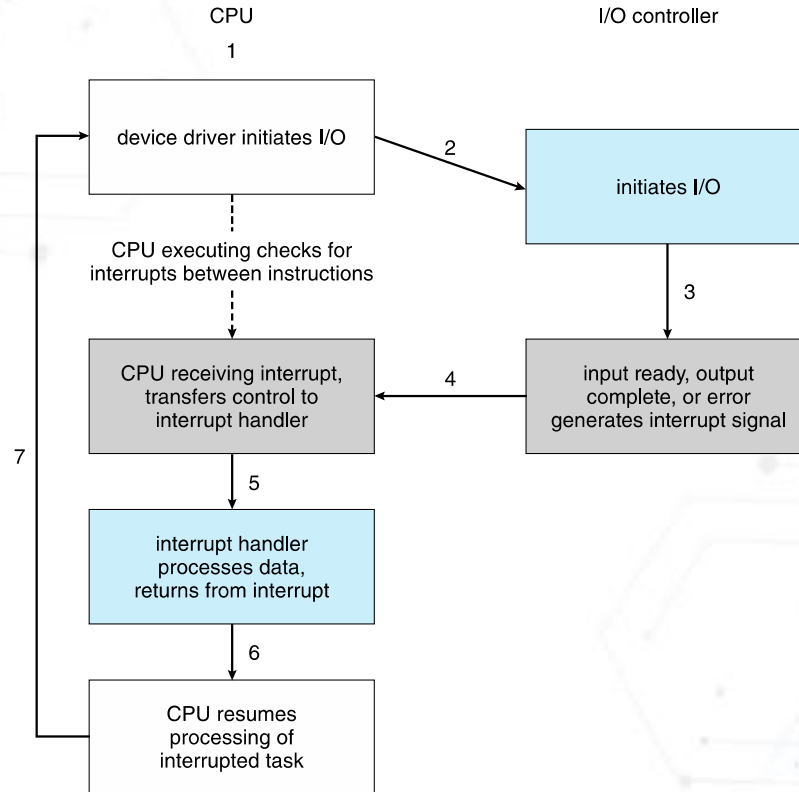


# Interrupt Handling



- Le système d'exploitation préserve l'état du CPU en stockant les registres et le compteur de programme.
- Il détermine le type d'interruption qui s'est produit
- des segments de code distincts déterminent l'action à entreprendre pour chaque type d'interruption.

# Interrupt-drive I/O Cycle



# I/O Structure



- Deux méthodes pour gérer les entrées/sorties (E/S) :
  - Après le début de l'E/S, le contrôle ne revient au programme utilisateur qu'après l'achèvement de l'E/S.
  - Après le début de l'E/S, le contrôle revient au programme utilisateur sans attendre l'achèvement de l'E/S.

## I/O Structure (Cont.)



- Après le début de l'E/S, le contrôle ne revient au programme utilisateur qu'après l'achèvement de l'E/S.
  - L'instruction d'attente (wait) met le CPU en veille jusqu'à la prochaine interruption.
  - La boucle d'attente (contention pour l'accès à la mémoire).
  - Au plus, une demande d'E/S est en cours à la fois, aucun traitement simultané d'E/S.
- Après le début de l'E/S, le contrôle revient au programme utilisateur sans attendre l'achèvement de l'E/S.
  - L'appel système - demande au système d'exploitation de permettre à l'utilisateur d'attendre la fin de l'E/S.
  - La table d'état des périphériques contient une entrée pour chaque périphérique d'E/S, indiquant son type, son adresse et son état.
  - Le système d'exploitation indexe dans la table des périphériques d'E/S pour déterminer l'état du périphérique et pour modifier l'entrée de la table pour inclure l'interruption.

# Computer Startup



- Le programme d'amorçage (bootstrap) est chargé lors de la mise sous tension ou du redémarrage.
  - Il est généralement stocké dans une mémoire morte (ROM) ou une mémoire morte programmable en lecture (EPROM), communément appelée firmware.
  - Il initialise tous les aspects du système,
  - charge le noyau du système d'exploitation et démarre l'exécution.




# Storage Structure

# Storage Structure

- Mémoire principale - seul support de stockage de grande capacité que le CPU peut accéder directement.
  - Accès aléatoire.
  - Généralement volatile.
  - Généralement une mémoire à accès aléatoire sous forme de mémoire dynamique à accès aléatoire (DRAM).
- Stockage secondaire - extension de la mémoire principale fournissant une grande capacité de stockage non volatile.

## Storage Structure (Cont.)

- 
- **Disques durs (HDD)** - plateaux rigides en métal ou en verre recouverts de matériau d'enregistrement magnétique.
    - La surface du disque est logiquement divisée en pistes, qui sont subdivisées en secteurs.
    - Le contrôleur de disque détermine l'interaction logique entre le dispositif et l'ordinateur.
  - **Dispositifs de mémoire non volatile (NVM)** - plus rapides que les disques durs, non volatils.
    - Diverses technologies.
    - Deviennent de plus en plus populaires à mesure que la capacité et les performances augmentent et que les prix baissent.



# Storage Definitions and Notation Review

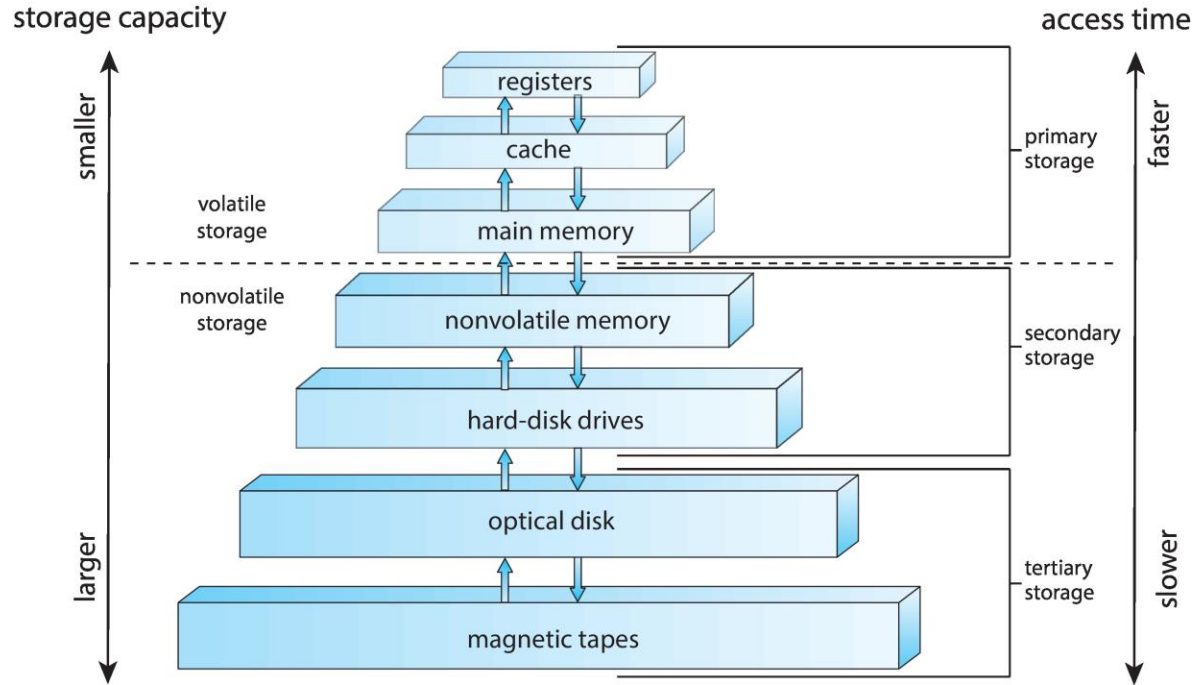
L'unité fondamentale de stockage informatique est le bit. Un bit peut contenir l'une des deux valeurs, 0 et 1. Tout autre stockage dans un ordinateur est basé sur des collections de bits. Avec suffisamment de bits, il est étonnant de voir combien de choses un ordinateur peut représenter : des nombres, des lettres, des images, des films, des sons, des documents et des programmes, pour n'en citer que quelques-uns. Un octet est composé de 8 bits, et sur la plupart des ordinateurs, c'est le plus petit morceau de stockage pratique. Par exemple, la plupart des ordinateurs n'ont pas d'instruction pour déplacer un bit, mais en ont une pour déplacer un octet. Un terme moins courant est le mot, qui est l'unité native de données d'une architecture informatique donnée. Un mot est constitué d'un ou plusieurs octets. Par exemple, un ordinateur qui a des registres de 64 bits et une adresse mémoire de 64 bits a généralement des mots de 64 bits (8 octets). Un ordinateur exécute de nombreuses opérations dans sa taille de mot native plutôt qu'un octet à la fois. Le stockage informatique, ainsi que la plupart du débit informatique, est généralement mesuré et manipulé en octets et en collections d'octets. Un kilo-octet, ou Ko, est de  $1\ 024$  octets ; un mégaoctet, ou Mo, est de  $1\ 024^2$  octets ; un gigaoctet, ou Go, est de  $1\ 024^3$  octets ; un téraoctet, ou To, est de  $1\ 024^4$  octets ; et un pétaoctet, ou Po, est de  $1\ 024^5$  octets. Les fabricants d'ordinateurs arrondissent souvent ces chiffres en disant qu'un mégaoctet est égal à 1 million d'octets et un gigaoctet est égal à 1 milliard d'octets. Les mesures de réseau sont une exception à cette règle générale ; elles sont données en bits (parce que les réseaux déplacent les données un bit à la fois).

# Storage Hierarchy

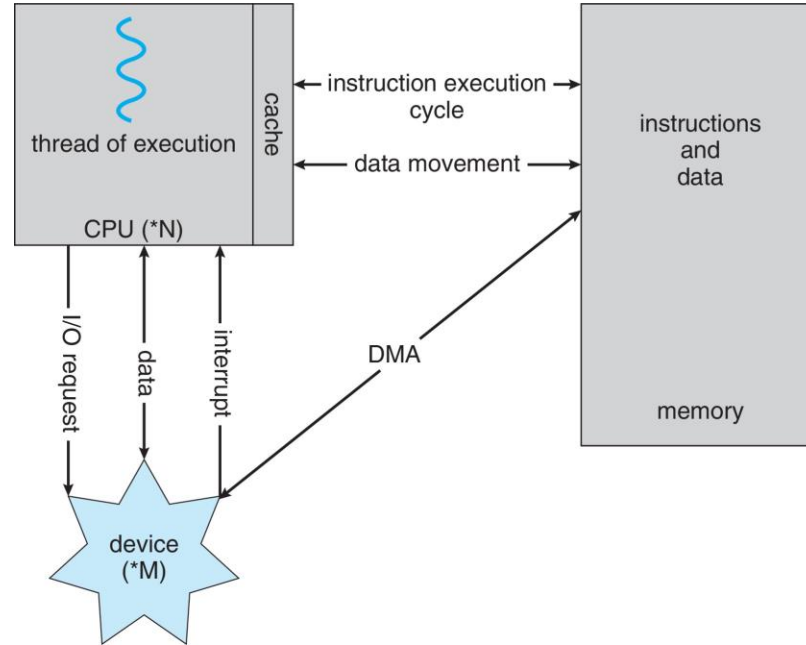


- Systèmes de stockage organisés en hiérarchie
  - Vitesse
  - Coût
  - Volatilité
- Mise en cache - copie des informations dans un système de stockage plus rapide ; la mémoire principale peut être considérée comme une mise en cache pour le stockage secondaire.
- Pilote de périphérique pour chaque contrôleur de périphérique pour gérer l'E/S
  - Fournit une interface uniforme entre le contrôleur et le noyau.

# Storage-Device Hierarchy



# How a Modern Computer Works



*A von Neumann architecture*

# Direct Memory Access Structure

- Utilisé pour les périphériques d'E/S haute vitesse capables de transmettre des informations à des vitesses proches de celles de la mémoire.
- Le contrôleur de périphérique transfère des blocs de données depuis le stockage tampon directement dans la mémoire principale sans intervention du CPU.
- Seule une interruption est générée par bloc, plutôt qu'une interruption par octet.

# Operating-System Operations

- Programme d'amorçage - code simple pour initialiser le système, charger le noyau
- Le noyau se charge
- Démarre les **démons système** (services fournis en dehors du noyau)
- Le noyau est **basé sur les interruptions** (matérielles et logicielles)
  - Interruption matérielle par l'un des périphériques
  - Interruption logicielle (**exception ou trappe**) :
    - Erreur logicielle (par exemple, division par zéro)
    - Demande de service du système d'exploitation - **appel système**
    - D'autres problèmes de processus comprennent la boucle infinie, les processus se modifiant mutuellement ou le système d'exploitation.
-

# Multiprogramming (Batch system)

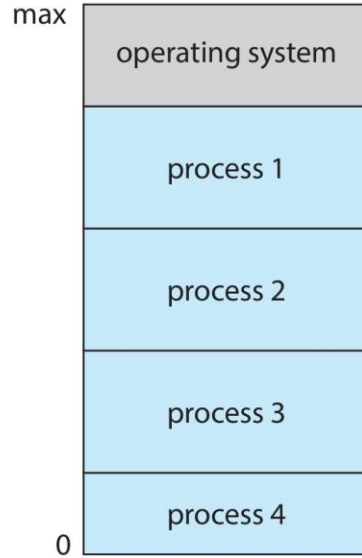
- Un seul utilisateur ne peut pas toujours occuper constamment le processeur et les périphériques d'E/S.
- La multiprogrammation organise les tâches (code et données) de sorte que le CPU a toujours une tâche à exécuter.
- Un sous-ensemble des tâches totales du système est conservé en mémoire.
- Une tâche est sélectionnée et exécutée via la planification des tâches.
- Lorsqu'une tâche doit attendre (par exemple, pour une E/S), le système d'exploitation passe à une autre tâche.
-

# Multitasking (Timesharing)

- Une extension logique des systèmes par lots - le CPU change de tâche si fréquemment que les utilisateurs peuvent interagir avec chaque tâche pendant son exécution, créant ainsi une informatique [interactive](#).
  - Le temps de réponse doit être inférieur à 1 seconde.
  - Chaque utilisateur a au moins un programme en cours d'exécution en mémoire ([processus](#)).
  - Si plusieurs tâches sont prêtes à être exécutées simultanément, cela nécessite une ordonnance d'unité centrale ([CPU scheduling](#)).
  - Si les processus ne rentrent pas en mémoire, le [remplacement](#) les déplace pour les exécuter.
  - [La mémoire virtuelle](#) permet l'exécution de processus qui ne sont pas entièrement en mémoire.



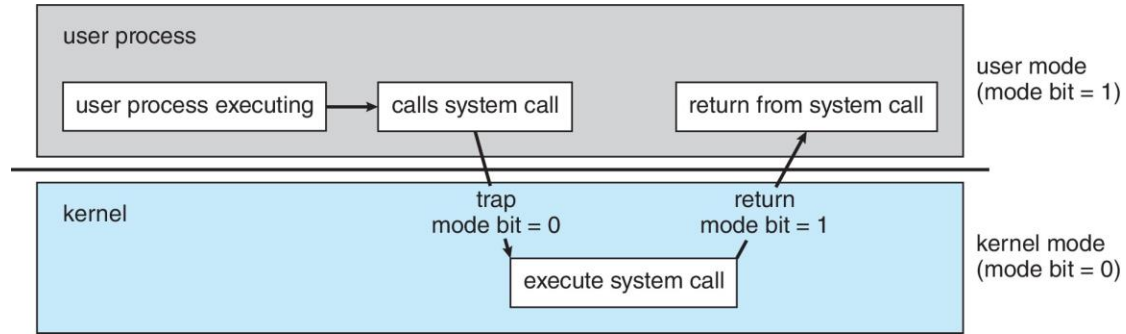
# Memory Layout for Multiprogrammed System



# Dual-mode Operation

- Le mode d'exploitation double permet au système d'exploitation de se protéger lui-même et les autres composants du système. Mode utilisateur et mode noyau.
- Le bit de mode est fourni par le matériel.
  - Il offre la possibilité de distinguer quand le système exécute du code utilisateur ou du code noyau.
  - Lorsqu'un utilisateur est en cours d'exécution, le bit de mode est défini sur "utilisateur".
  - Lorsque le code du noyau est en cours d'exécution, le bit de mode est défini sur "noyau".
- Comment garantir que l'utilisateur ne définit pas explicitement le bit de mode sur "noyau" ?
  - L'appel système change le mode en mode noyau, le retour de l'appel le réinitialise en mode utilisateur.
- Certaines instructions sont désignées comme privilégiées, exécutables uniquement en mode noyau.

# Transition from User to Kernel Mode



# Timer

- Utilisation d'une minuterie pour prévenir les boucles infinies (ou les processus accaparant des ressources)
  - La minuterie est réglée pour interrompre l'ordinateur après une certaine période de temps.
  - Maintien d'un compteur décrémente par l'horloge physique.
  - Le système d'exploitation configure le compteur (instruction privilégiée).
  - Lorsque le compteur atteint zéro, une interruption est générée.
  - Configuration effectuée avant la planification du processus pour reprendre le contrôle ou terminer un programme qui dépasse le temps alloué.

# Process Management

- Un processus est un programme en cours d'exécution. C'est une unité de travail au sein du système. Un programme est une entité passive ; un processus est une entité active.
- Le processus a besoin de ressources pour accomplir sa tâche
  - UCP, mémoire, E/S, fichiers
  - Initialisation des données La terminaison d'un processus nécessite la récupération de toutes les ressources réutilisables
- Un processus à thread unique a un **compteur de programme** spécifiant l'emplacement de l'instruction suivante à exécuter
  - Le processus exécute les instructions séquentiellement, une à la fois, jusqu'à la fin
- Un processus multithread a un compteur de programme par thread
- Typiquement, le système a de nombreux processus, certains utilisateur, certains du système d'exploitation s'exécutant simultanément sur un ou plusieurs processeurs
  - La concurrence se fait en multiplexant les processeurs entre les processus/threads

# Process Management Activities

Le système d'exploitation est responsable des activités suivantes liées à la gestion des processus :

- Création et suppression des processus utilisateur et système
- Suspension et reprise des processus
- Fourniture de mécanismes de synchronisation des processus
- Fourniture de mécanismes de communication entre processus
- Fourniture de mécanismes de gestion des impasses (deadlock)

# Memory Management

- Pour exécuter un programme, toutes (ou une partie) des instructions doivent être en mémoire,
- de même que toutes (ou une partie) des données nécessaires au programme.
  - La gestion de la mémoire détermine ce qui est en mémoire et quand, optimiser l'utilisation du CPU et la réponse de l'ordinateur aux utilisateurs.
- Les activités de gestion de la mémoire comprennent
  - le suivi des parties de la mémoire actuellement utilisées et par qui,
  - la décision des processus (ou de parties de ceux-ci) et des données à déplacer dans et hors de la mémoire, ainsi que
  - l'allocation et la désallocation d'espace mémoire selon les besoins.

# File-system Management

- Le système d'exploitation offre une vue uniforme et logique du stockage de l'information,
  - abstrayant les propriétés physiques vers une unité de stockage logique appelée fichier.
  - Chaque support est contrôlé par un périphérique (par exemple, un lecteur de disque, un lecteur de bande),
    - avec des propriétés variables telles que la vitesse d'accès, la capacité, le taux de transfert de données et la méthode d'accès (séquentiel ou aléatoire).
- La gestion du système de fichiers implique généralement
  - l'organisation des fichiers en répertoires,
  - avec des contrôles d'accès pour déterminer qui peut accéder à quoi.
  - Les activités du système d'exploitation comprennent
    - la création et la suppression de fichiers et de répertoires,
    - des primitives pour manipuler les fichiers et les répertoires,
    - le mappage des fichiers sur un stockage secondaire,
    - ainsi que la sauvegarde des fichiers sur un support de stockage stable (non volatile).



# Mass-Storage Management

- Généralement, les disques sont utilisés pour stocker des données qui ne rentrent pas dans la mémoire principale ou des données qui doivent être conservées pendant une "longue" période.
- Une gestion appropriée est d'une importance cruciale, car la vitesse globale des opérations informatiques dépend du sous-système de disque et de ses algorithmes.
- Les activités du système d'exploitation comprennent
  - le montage et le démontage,
  - la gestion de l'espace libre,
  - l'allocation de stockage,
  - la planification des disques,
  - le partitionnement et
  - la protection.

# Caching

- Principe important, réalisé à de nombreux niveaux dans un ordinateur (matériel, système d'exploitation, logiciel).
- Les informations en cours d'utilisation sont copiées temporairement de la mémoire plus lente vers une mémoire plus rapide.
- La mémoire plus rapide (cache) est d'abord vérifiée pour déterminer si les informations s'y trouvent.
  - Si c'est le cas, les informations sont utilisées directement depuis le cache (rapide).
  - Sinon, les données sont copiées dans le cache et utilisées à partir de là.
- Le cache est plus petit que la mémoire qu'il met en cache, et
  - la gestion du cache est un problème de conception important,
  - incluant la taille du cache et la politique de remplacement.

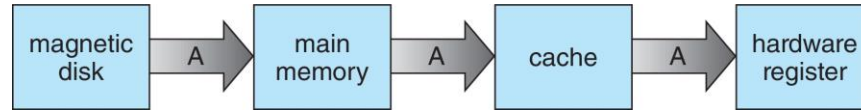
# Characteristics of Various Types of Storage

Level	1	2	3	4	5
Name	registers	cache	main memory	solid-state disk	magnetic disk
Typical size	< 1 KB	< 16MB	< 64GB	< 1 TB	< 10 TB
Implementation technology	custom memory with multiple ports CMOS	on-chip or off-chip CMOS SRAM	CMOS SRAM	flash memory	magnetic disk
Access time (ns)	0.25-0.5	0.5-25	80-250	25,000-50,000	5,000,000
Bandwidth (MB/sec)	20,000-100,000	5,000-10,000	1,000-5,000	500	20-150
Managed by	compiler	hardware	operating system	operating system	operating system
Backed by	cache	main memory	disk	disk	disk or tape

implicit

# Migration of data "A" from Disk to Register

- Les environnements multitâches doivent veiller à utiliser la valeur la plus récente, peu importe où elle est stockée dans la hiérarchie de stockage.



- Dans un environnement multiprocesseur, il est nécessaire de garantir la cohérence du cache matériel de manière à ce que tous les processeurs aient la valeur la plus récente dans leur cache.
- Dans un environnement distribué, la situation est encore plus complexe, car
  - plusieurs copies d'une donnée peuvent coexister.
  - Diverses solutions à ce problème sont abordées dans le chapitre 19.

# I/O Subsystem

- L'un des objectifs du système d'exploitation est de masquer les particularités des périphériques matériels à l'utilisateur.
- Le sous-système d'E/S est responsable de
  - la gestion de la mémoire d'E/S, y compris le tampon (stockage temporaire des données pendant leur transfert), le cache (stockage de parties des données dans une mémoire plus rapide pour améliorer les performances) et le spouleur (le chevauchement de la sortie d'un travail avec l'entrée d'autres travaux).
  - Il offre une interface générale pour les pilotes de périphériques.
  - Il inclut également des pilotes spécifiques pour les périphériques matériels particuliers.

# Protection and Security

- **Protection** – tout mécanisme permettant de contrôler l'accès des processus ou des utilisateurs aux ressources définies par le système d'exploitation.
- **Sécurité** – défense du système contre les attaques internes et externes.
  - Cela inclut une vaste gamme de menaces telles que le déni de service, les vers, les virus, le vol d'identité et le vol de services.
- Les systèmes font généralement une distinction entre les utilisateurs pour déterminer qui peut faire quoi.
  - Les identités des utilisateurs (**identifiants d'utilisateur**, identifiants de sécurité) comprennent le nom et le numéro associé, un par utilisateur.
  - L'identifiant d'utilisateur est ensuite associé à tous les fichiers, processus de cet utilisateur pour déterminer le contrôle d'accès.
  - L'identifiant de groupe (**ID de groupe**) permet de définir un ensemble d'utilisateurs et de gérer les contrôles, puis est également associé à chaque processus et fichier.
  - **L'élévation de privilèges** permet à un utilisateur de passer à un ID effectif avec plus de droits.

# Virtualization

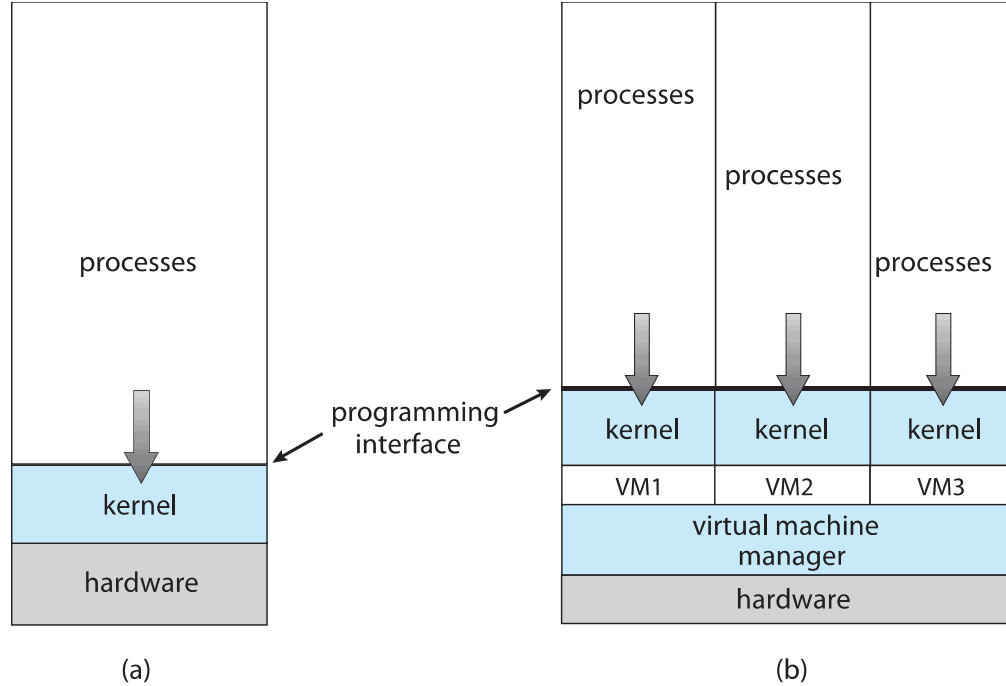
- Permet aux systèmes d'exploitation d'exécuter des applications au sein d'autres systèmes d'exploitation.
  - Industrie vaste et en expansion.
- L'**émulation** est utilisée lorsque le type de CPU source est différent du type cible (par exemple, de PowerPC à Intel x86).
  - Généralement, c'est la méthode la plus lente.
  - Lorsque le langage informatique n'est pas compilé en code natif - **Interprétation**.
- **Virtualisation** - le système d'exploitation est nativement compilé pour le CPU, exécutant des systèmes d'exploitation **invités** également nativement compilés.
  - Considérez VMware exécutant des invités WinXP, chacun exécutant des applications, le tout sur un **hôte** natif WinXP.
  - **VMM** (Gestionnaire de machine virtuelle) fournit des services de virtualisation.

# Virtualization (cont.)

- Les cas d'utilisation impliquent des ordinateurs portables et des ordinateurs de bureau exécutant plusieurs systèmes d'exploitation pour l'exploration ou la compatibilité.
  - Un ordinateur portable Apple exécutant Mac OS X en tant qu'hôte, Windows en tant qu'invité.
  - Développement d'applications pour plusieurs systèmes d'exploitation sans avoir besoin de plusieurs systèmes.
  - Tests d'assurance qualité d'applications sans avoir besoin de plusieurs systèmes.
  - Exécution et gestion d'environnements de calcul dans des centres de données.
- Les VMM peuvent s'exécuter nativement, auquel cas ils sont également l'hôte.
  - Il n'y a pas d'hôte polyvalent alors (VMware ESX et Citrix XenServer).



# Computing Environments - Virtualization



# Distributed Systems

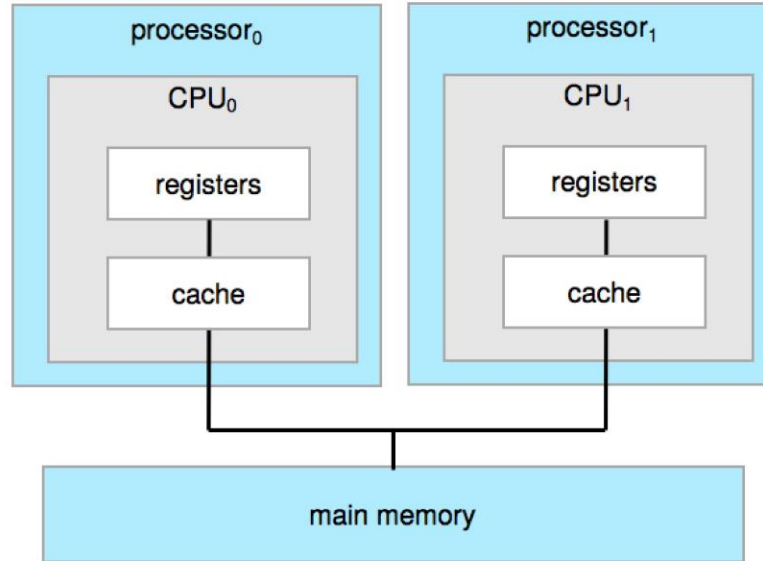
- Ensemble de systèmes distincts, éventuellement hétérogènes, interconnectés.
  - Le réseau est un chemin de communication, TCP/IP étant le plus courant.
    - Réseau local (LAN)
    - Réseau étendu (WAN)
    - Réseau métropolitain (MAN)
    - Réseau personnel (PAN)
- Le système d'exploitation réseau fournit des fonctionnalités entre les systèmes à travers le réseau.
  - Le schéma de communication permet aux systèmes d'échanger des messages.
  - Illusion d'un seul système.

# **Computer System Architecture**

# Computer-System Architecture

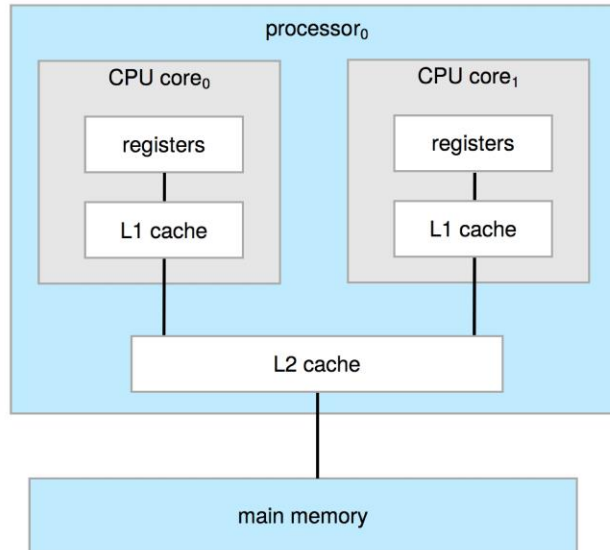
- La plupart des systèmes utilisent un seul processeur polyvalent. De plus, la plupart des systèmes disposent également de processeurs spécialisés.
- Les systèmes multiprocesseurs sont de plus en plus utilisés et importants.
  - Ils sont également connus sous le nom de systèmes parallèles, ou systèmes fortement couplés.
  - Les avantages incluent
    1. une augmentation du débit,
    2. une économie d'échelle,
    3. une fiabilité accrue avec une dégradation progressive ou une tolérance aux pannes.
  - Il existe deux types :
    1. Multiprocessing asymétrique - chaque processeur est affecté à une tâche spécifique.
    2. Multiprocessing symétrique - chaque processeur effectue toutes les tâches.

# Symmetric Multiprocessing Architecture

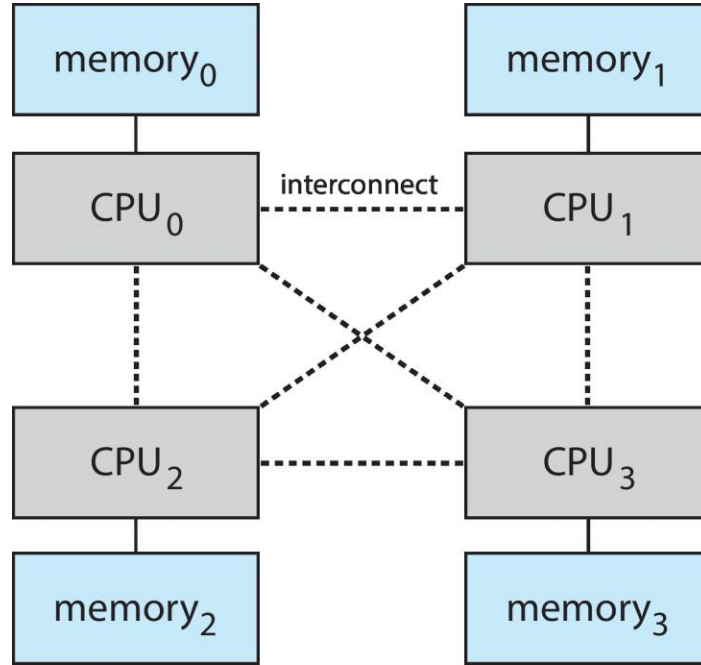


# Dual-Core Design

- Multi-puce et multicœur
- Systèmes contenant toutes les puces
  - Châssis contenant plusieurs systèmes séparés



# Non-Uniform Memory Access System

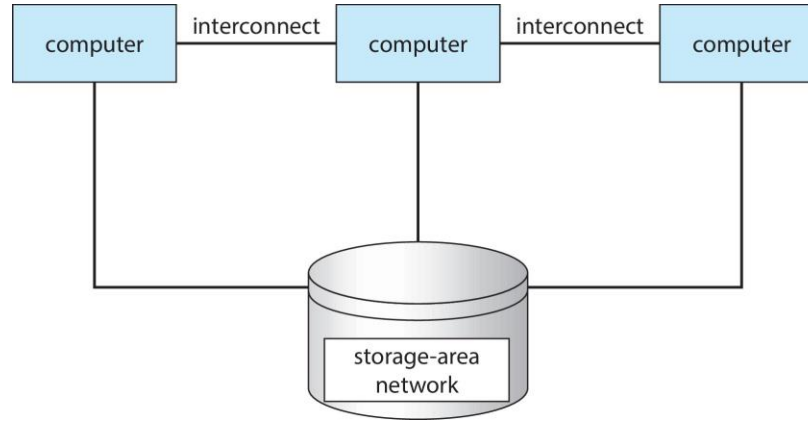


# Clustered Systems

- Comme les systèmes multiprocesseurs, mais plusieurs systèmes travaillant ensemble
  - Partage généralement du stockage via un réseau de zone de stockage (SAN)
  - Fournit un service à haute disponibilité qui survit aux défaillances
    - Le clustering asymétrique a une machine en mode veille active
    - Le clustering symétrique a plusieurs nœuds exécutant des applications, se surveillant mutuellement
  - Certains clusters sont destinés à la haute performance informatique (HPC)
    - Les applications doivent être écrites pour utiliser la parallélisation
  - Certains ont un gestionnaire de verrous distribué (DLM) pour éviter les opérations conflictuelles

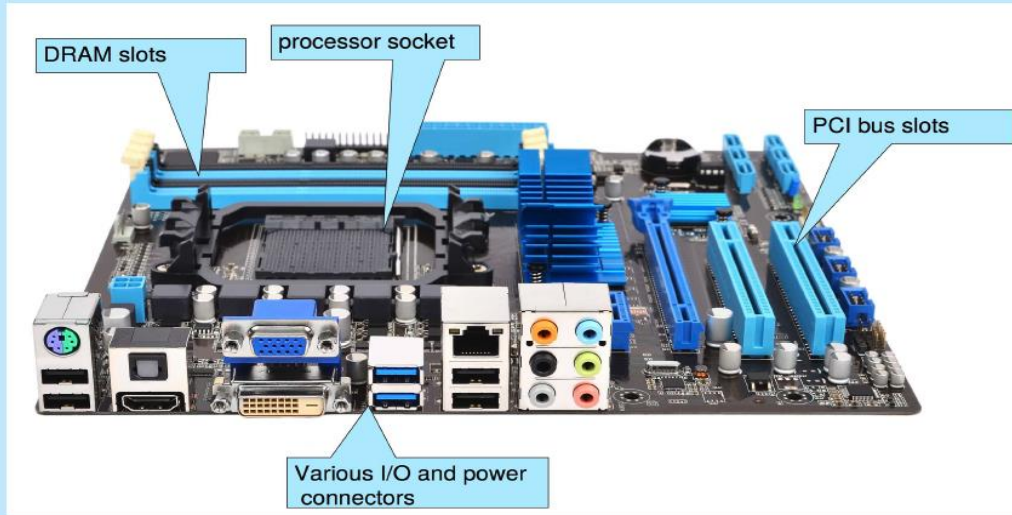


# Clustered Systems



# PC Motherboard

Consider the desktop PC motherboard with a processor socket shown below:



This board is a fully-functioning computer, once its slots are populated. It consists of a processor socket containing a CPU, DRAM sockets, PCIe bus slots, and I/O connectors of various types. Even the lowest-cost general-purpose CPU contains multiple cores. Some motherboards contain multiple processor sockets. More advanced computers allow more than one system board, creating NUMA systems.

# Computer System Environments

# Computing Environments

- Traditionnel
- Mobile
- Client-serveur
- Pair-à-pair
- Informatique en nuage (Cloud computing)
- Temps réel Embarqué

# Traditional

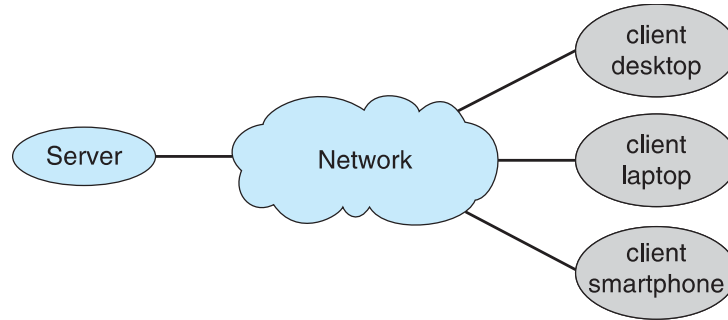
- Machines autonomes à usage général
- Mais de plus en plus floues car la plupart des systèmes interagissent les uns avec les autres (c'est-à-dire, l'Internet)
- Les **portails** offrent un accès web aux systèmes internes
- Les **ordinateurs en réseau (clients légers)** sont comme des terminaux Web
- Les ordinateurs portables interagissent via **des réseaux sans fil**
- La mise en réseau devient omniprésente - même les systèmes domestiques utilisent **des pare-feu** pour protéger les ordinateurs domestiques contre les attaques Internet

# Mobile

- Smartphones portables, tablettes, etc.
- Quelle est la différence fonctionnelle entre eux et un ordinateur portable "traditionnel" ?
- Fonctionnalités supplémentaires - plus de fonctionnalités OS (GPS, gyroscope)
- Permet de nouveaux types d'applications comme la réalité augmentée
- Utilise le sans fil IEEE 802.11 ou les réseaux de données cellulaires pour la connectivité
- Les leaders sont [Apple iOS](#) et [Google Android](#)

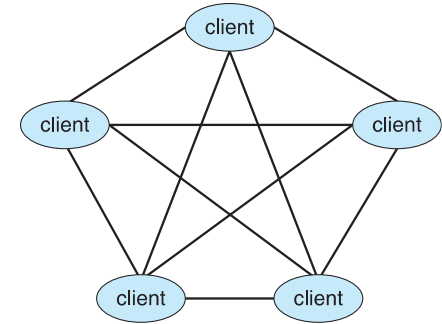
# Client Server

- Informatique client-serveur
  - Les terminaux passifs ont été remplacés par des PC intelligents
  - De nombreux systèmes sont désormais des serveurs, répondant aux demandes générées par les clients
    - ▶ Un **système client-serveur** fournit une interface au client pour demander des services (par exemple, une base de données)
    - ▶ Un **système de serveur de fichiers** fournit une interface pour que les clients stockent et récupèrent des fichiers



# Peer-to-Peer

- Un autre modèle de système distribué
- P2P ne fait pas la distinction entre clients et serveurs
  - Au lieu de cela, tous les nœuds sont considérés comme des pairs
  - Chacun peut agir en tant que client, serveur ou les deux
  - Le nœud doit rejoindre le réseau P2P
    - Enregistre son service auprès du service de recherche central sur le réseau, ou
    - Diffuse une demande de service et répond aux demandes de service via un protocole de découverte
  - Des exemples incluent Napster et Gnutella, la [voix sur IP \(VoIP\)](#) telle que Skype





# Cloud Computing

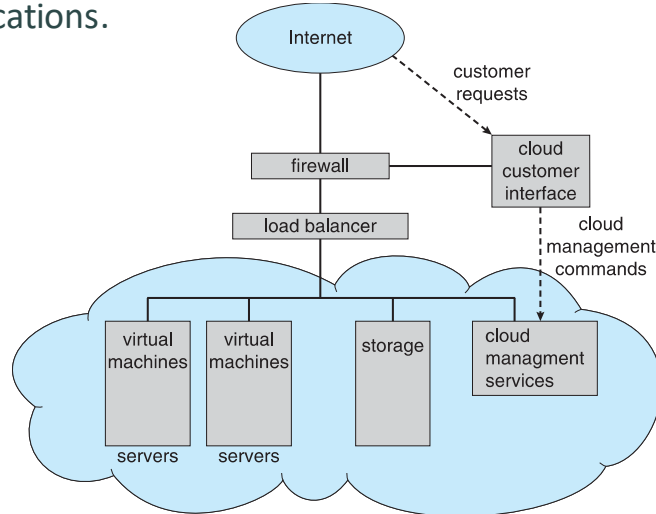
- Fournit des services informatiques, de stockage, voire des applications en tant que service à travers un réseau
- Extension logique de la virtualisation car elle utilise la virtualisation comme base pour sa fonctionnalité.
  - Amazon [EC2](#) dispose de milliers de serveurs, de millions de machines virtuelles, de pétaoctets de stockage disponibles sur Internet, avec une tarification basée sur l'utilisation

# Cloud Computing (Cont.)

- De nombreux types
  - **Cloud public** : accessible via Internet pour toute personne prête à payer
  - **Cloud privé** : géré par une entreprise pour son propre usage
  - **Cloud hybride** : inclut à la fois des composants de cloud public et privé
  - Logiciel en tant que service (**SaaS**) : une ou plusieurs applications disponibles via Internet (par exemple, un traitement de texte)
  - Plateforme en tant que service (**PaaS**) : pile logicielle prête à être utilisée par une application via Internet (par exemple, un serveur de base de données)
  - Infrastructure en tant que service (**IaaS**) : serveurs ou stockage disponibles sur Internet (par exemple, stockage disponible pour des utilisations de sauvegarde)

# Cloud Computing (cont.)

- Les environnements de cloud computing sont composés de systèmes d'exploitation traditionnels, de machines virtuelles, ainsi que d'outils de gestion du cloud.
  - La connectivité Internet nécessite des mesures de sécurité telles que des pare-feu.
  - Des équilibres de charge répartissent le trafic entre plusieurs applications.



# Real-Time Embedded Systems

- Les systèmes embarqués en temps réel constituent la forme la plus répandue d'ordinateurs.
  - Ils varient considérablement, allant des systèmes spéciaux à usage limité aux **systèmes d'exploitation en temps réel**.
  - Leur utilisation est en expansion, et
- il existe de nombreux autres environnements informatiques spéciaux.
  - Certains disposent de systèmes d'exploitation, tandis que d'autres effectuent des tâches sans système d'exploitation.
- Un système d'exploitation en temps réel impose des contraintes temporelles bien définies,
  - exigeant que le traitement soit effectué dans ces limites.
  - Un fonctionnement correct n'est assuré que si ces contraintes sont respectées.

# Free and Open-Source Operating Systems

- Les systèmes d'exploitation rendus disponibles sous forme de code source plutôt que sous forme binaire fermée et propriétaire.
- Ils vont à l'encontre du mouvement de protection contre la copie et de gestion des droits numériques (DRM).
- Cette approche a été initiée par la Free Software Foundation (FSF), qui propose la licence GNU General Public License (GPL) avec la philosophie du logiciel libre.
  - Le logiciel libre et le logiciel open source sont deux idées différentes défendues par des groupes de personnes différents
    - (<https://www.gnu.org/philosophy/open-source-misses-the-point.en.html>).
- Des exemples incluent GNU/Linux et BSD UNIX (y compris le noyau de Mac OS X), parmi de nombreux autres.
- On peut utiliser des machines virtuelles telles que VMware Player (gratuit sous Windows) ou Virtualbox (open source et gratuit sur de nombreuses plateformes - <http://www.virtualbox.com>)
  - pour exécuter des systèmes d'exploitation invités à des fins d'exploration.

# The Study of Operating Systems

- ❖ Il n'y a jamais eu de moment plus intéressant pour étudier les systèmes d'exploitation, et cela n'a jamais été aussi facile. Le mouvement open source a pris le dessus sur les systèmes d'exploitation, les rendant disponibles sous forme de code source et binaire (exécutable). La liste des systèmes d'exploitation disponibles dans les deux formats comprend Linux, BSD UNIX, Solaris et une partie de macOS. La disponibilité du code source nous permet d'étudier les systèmes d'exploitation de l'intérieur. Des questions auxquelles nous pouvions autrefois répondre uniquement en consultant la documentation ou le comportement d'un système d'exploitation peuvent désormais être résolues en examinant le code lui-même.
- ❖ Les systèmes d'exploitation qui ne sont plus commercialement viables ont également été libérés en open source, nous permettant d'étudier le fonctionnement des systèmes à une époque où les ressources en CPU, mémoire et stockage étaient plus limitées. Une liste étendue mais incomplète de projets de systèmes d'exploitation open source est disponible sur [https://curlie.org/Computers/Software/Operating\\_Systems/Open\\_Source/](https://curlie.org/Computers/Software/Operating_Systems/Open_Source/).
- ❖ De plus, l'essor de la virtualisation en tant que fonction informatique courante (et fréquemment gratuite) permet d'exécuter de nombreux systèmes d'exploitation sur un système central. Par exemple, VMware (<http://www.vmware.com>) propose un "lecteur" gratuit pour Windows sur lequel des centaines de "machines virtuelles" gratuites peuvent être exécutées. Virtualbox (<http://www.virtualbox.com>) propose un gestionnaire de machine virtuelle gratuit et open source sur de nombreuses plates-formes. En utilisant de tels outils, les étudiants peuvent essayer des centaines de systèmes d'exploitation sans avoir besoin de matériel dédié.
- ❖ L'avènement des systèmes d'exploitation open source a également facilité la transition de l'étudiant au développeur de systèmes d'exploitation. Avec un peu de connaissance, d'efforts et une connexion Internet, un étudiant peut même créer une nouvelle distribution de système d'exploitation. Il y a quelques années à peine, il était difficile, voire impossible, d'obtenir un accès au code source. Maintenant, un tel accès est limité uniquement par l'intérêt, le temps et l'espace disque dont dispose un étudiant.

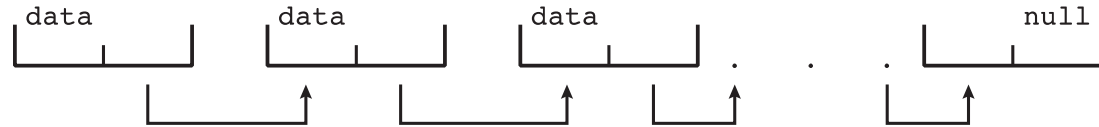


# **Kernel Data Structure**

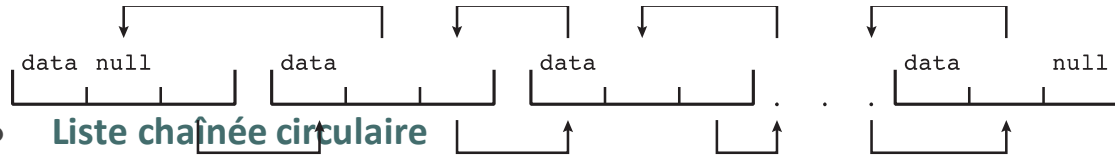
# Kernel Data Structures

- Plusieurs structures de données similaires aux structures de données de programmation standard

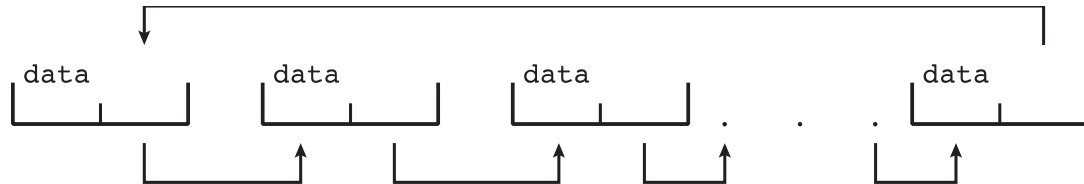
- **Liste chaînée simple**



- **Liste chaînée double**



- **Liste chaînée circulaire**



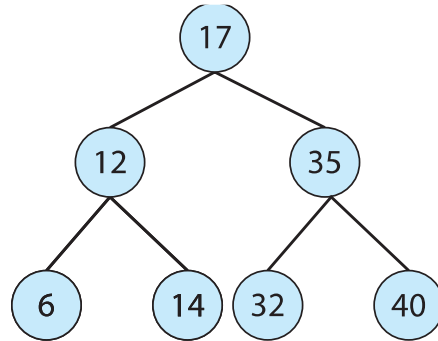


# Kernel Data Structures

- **Arbre binaire de recherche**

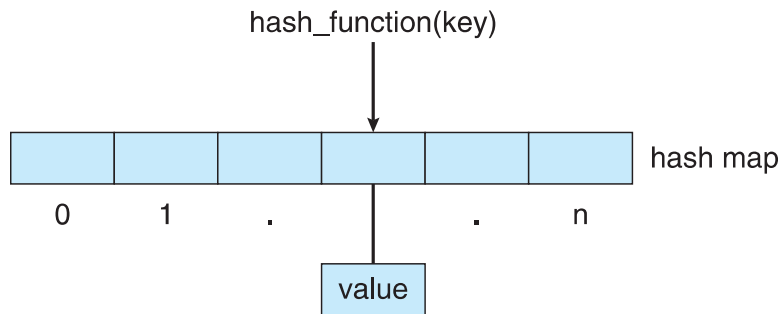
left <= rightgauche <= droite

- La performance de recherche est  $O(n)$
- **Un arbre binaire de recherche équilibré est  $O(\lg n)$**



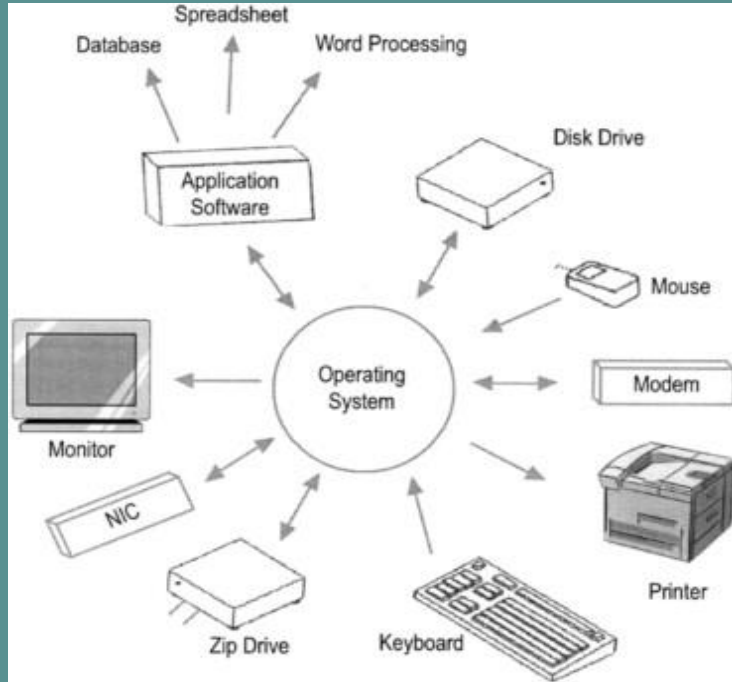
# Kernel Data Structures

- **Fonction de hachage** peut créer une **table de hachage**



- **Bitmap** –chaîne de n chiffres binaires représentant l'état de n éléments
- Les structures de données Linux sont définies dans les fichiers d'en-tête `<linux/list.h>`, `<linux/kfifo.h>`, `<linux/rbtree.h>`

# Operating Systems



Prof. B. A. Bagula  
Department of Computer Science,  
University of the Western Cape, South Africa.