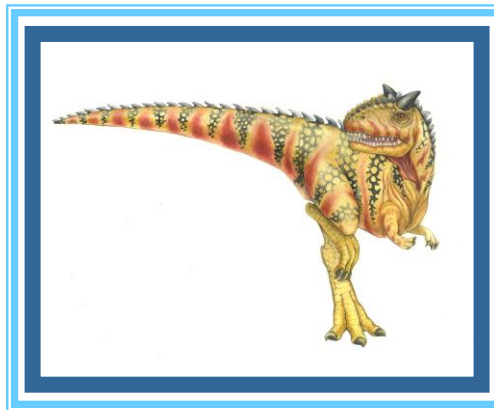
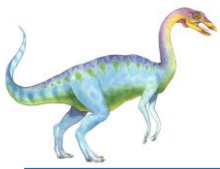


Chapter 1: Introduction





Chapter 1: Introduction

- Les systèmes d'exploitation ?
- Organisation des systèmes informatiques
- Architecture des systèmes informatiques
- Opérations des systèmes d'exploitation
- Gestion des ressources
- Sécurité et protection
- Virtualisation
- Systèmes distribués
- Structures de données du noyau
- Environnements informatiques
- Systèmes d'exploitation libres 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 libres et open source

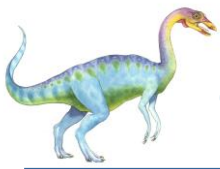




Que signifie le terme système d'exploitation ?

- Un système d'exploitation consiste à « remplissez les blancs »
- Qu'en est-il de :
 - Voiture
 - Avion
 - Imprimante
 - Machine à laver
 - Grille-pain
 - Compilateur
 - Etc.





Qu'est-ce qu'un système d'exploitation ?

- Un **programme** qui agit comme **intermédiaire** entre **l'utilisateur** d'un ordinateur et le **matériel** informatique
- **Objectifs du système d'exploitation :**
 1. **Exécuter les programmes utilisateur** et faciliter la résolution des problèmes des utilisateurs
 2. **Rendre le système** informatique **facile à utiliser**
 3. **Utiliser le matériel** informatique de **manière efficace**

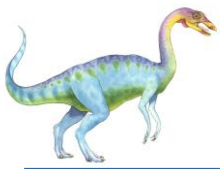




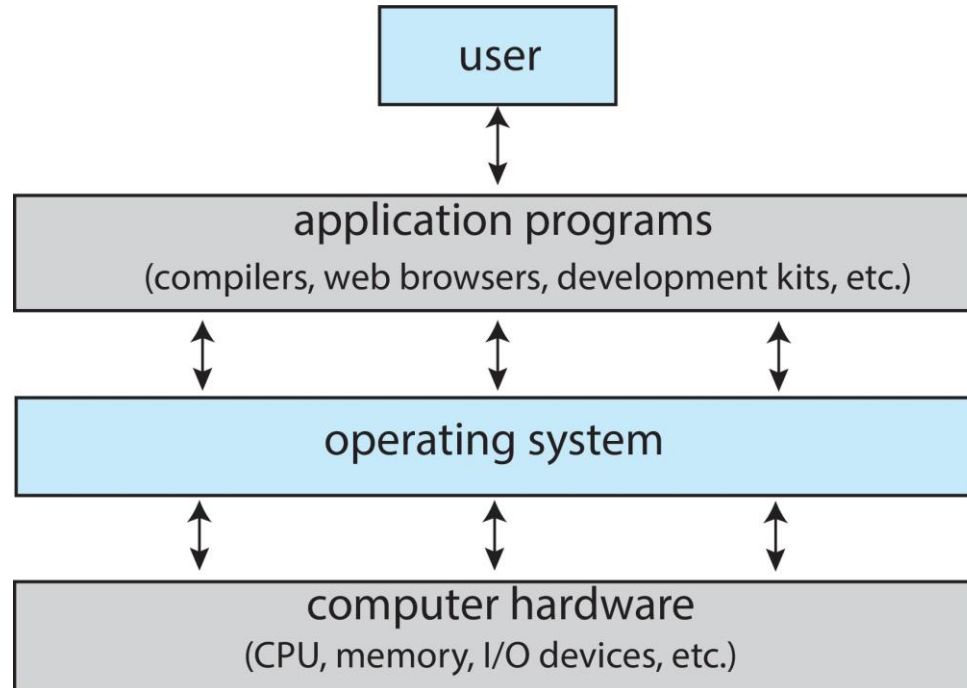
Structure du système informatique

- Le système informatique peut être divisé en quatre composants :
 1. **Matériel** : fournit les ressources informatiques de base
 - ▶ CPU, mémoire, périphériques d'E/S
 2. **Système d'exploitation**
 - ▶ Contrôle et coordonne l'utilisation du matériel entre diverses applications et utilisateurs
 3. **Programmes d'application** : définissent les façons dont les ressources du système sont utilisées pour résoudre les problèmes informatiques des utilisateurs
 - ▶ Traitements de texte, compilateurs, navigateurs Web, systèmes de base de données, jeux vidéo
 4. **Utilisateurs**
 - ▶ Personnes, machines, autres ordinateurs





Vue abstraite des composants d'un ordinateur

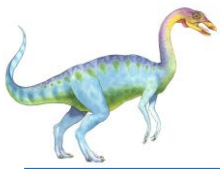




À quoi servent les systèmes d'exploitation

- Cela dépend du point de vue
- Les utilisateurs veulent de la **commodité**, une **facilité d'utilisation** et de bonnes **performances**
 - Ils ne se soucient pas de **l'utilisation des ressources**
- Un **ordinateur central** ou un **mini-ordinateur** doit répondre aux besoins de tous les utilisateurs
 - Le système d'exploitation gère les ressources, contrôle le matériel et exécute les programmes utilisateur.
- Les utilisateurs de systèmes dédiés tels que les **workstations** ont des ressources dédiées mais utilisent fréquemment des ressources partagées à partir de **servers**
- Les **appareils mobiles** tels que les **smartphones** et les tablettes sont pauvres en ressources, optimisés pour la convivialité et la durée de vie de la batterie
 - Interfaces utilisateur mobiles telles que les écrans tactiles, la reconnaissance vocale
- Certains ordinateurs ont peu ou pas d'interface utilisateur, comme les **ordinateurs intégrés** dans les appareils et les automobiles
 - Fonctionnent principalement sans intervention de l'utilisateur





Définition des systèmes d'exploitation

- Le terme OS recouvre de nombreux rôles (SE)
 - En raison de la multitude de conceptions et d'utilisations des systèmes d'exploitation
 - Présent dans les grille-pains, les navires, les engins spatiaux, les consoles de jeux, les téléviseurs et les systèmes de contrôle industriels
 - Né lorsque les ordinateurs à usage fixe pour l'armée sont devenus plus polyvalents et ont nécessité une gestion des ressources et un contrôle des programmes

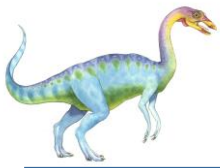




Définition des systèmes d'exploitation

- Il n'existe pas de définition universellement.
- Un système d'exploitation est **un ensemble de programmes** assurant la gestion des ressources matérielles et logicielles, l'ordonnancement des processus et l'interfaçage entre l'utilisateur et la machine.
- Tout le reste est soit :
 - Un **programme système** (fourni avec le système d'exploitation, mais ne faisant pas partie du noyau), soit
 - Un **programme applicatif**, c'est-à-dire tout programme non associé au système d'exploitation.
- Les systèmes d'exploitation modernes, destinés aux ordinateurs à usage général et aux appareils mobiles, incluent également des **middleware** un ensemble de **frameworks logiciels** qui offrent des services supplémentaires aux développeurs d'applications, tels que les bases de données, le multimédia et les graphiques.





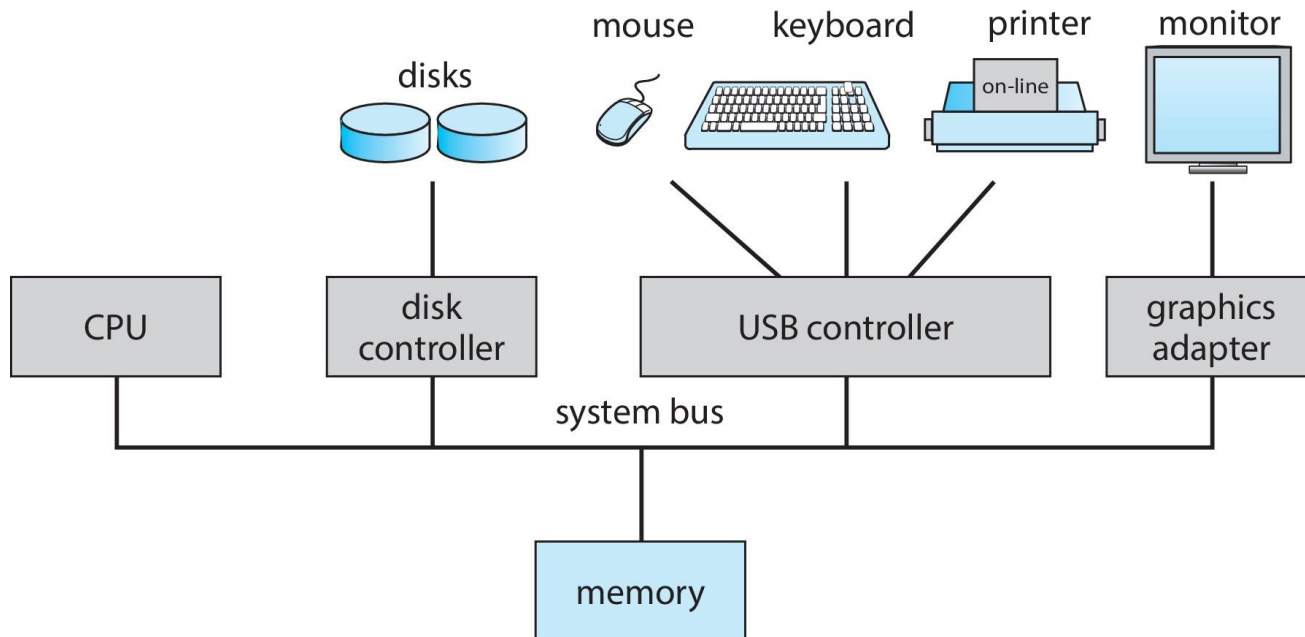
Aperçu de la structure du système informatique

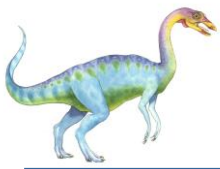




Organisation du système informatique

- Fonctionnement d'un système informatique
 - Un ou plusieurs CPUs, et de contrôleurs de périphériques connectent via un **bus** commun permettant d'accéder à la mémoire partagée
 - Exécution simultanée de CPUs et de périphériques en compétition pour les cycles de mémoire





Fonctionnement du système informatique

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

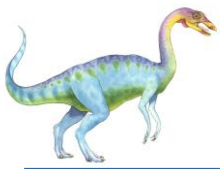




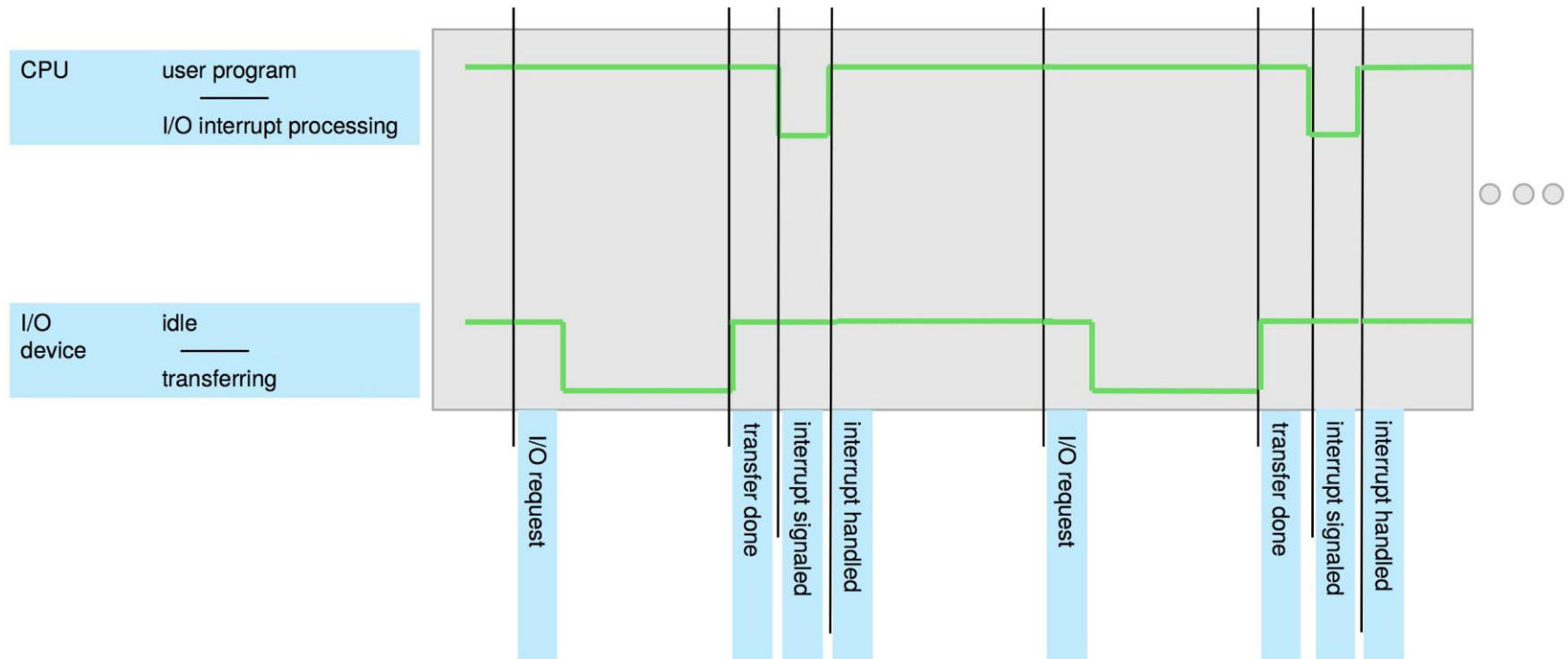
Fonctions communes des interruptions

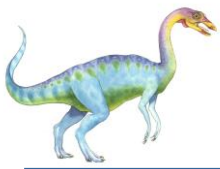
- Une interruption transfère le contrôle à la routine de service d'interruption, généralement via **le vecteur d'interruption**, qui contient les adresses de toutes les routines de service.
- L'architecture d'interruption doit enregistrer l'adresse de l'instruction interrompue
- Une **interruption** ou **exception** est un **signal** logiciel généré soit par une erreur, soit par une demande de l'utilisateur.
- Un système d'exploitation est **piloté par interruption**





Interrupt Timeline

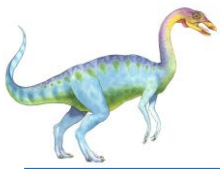




Interrupt Handling (Gestion des interruptions)

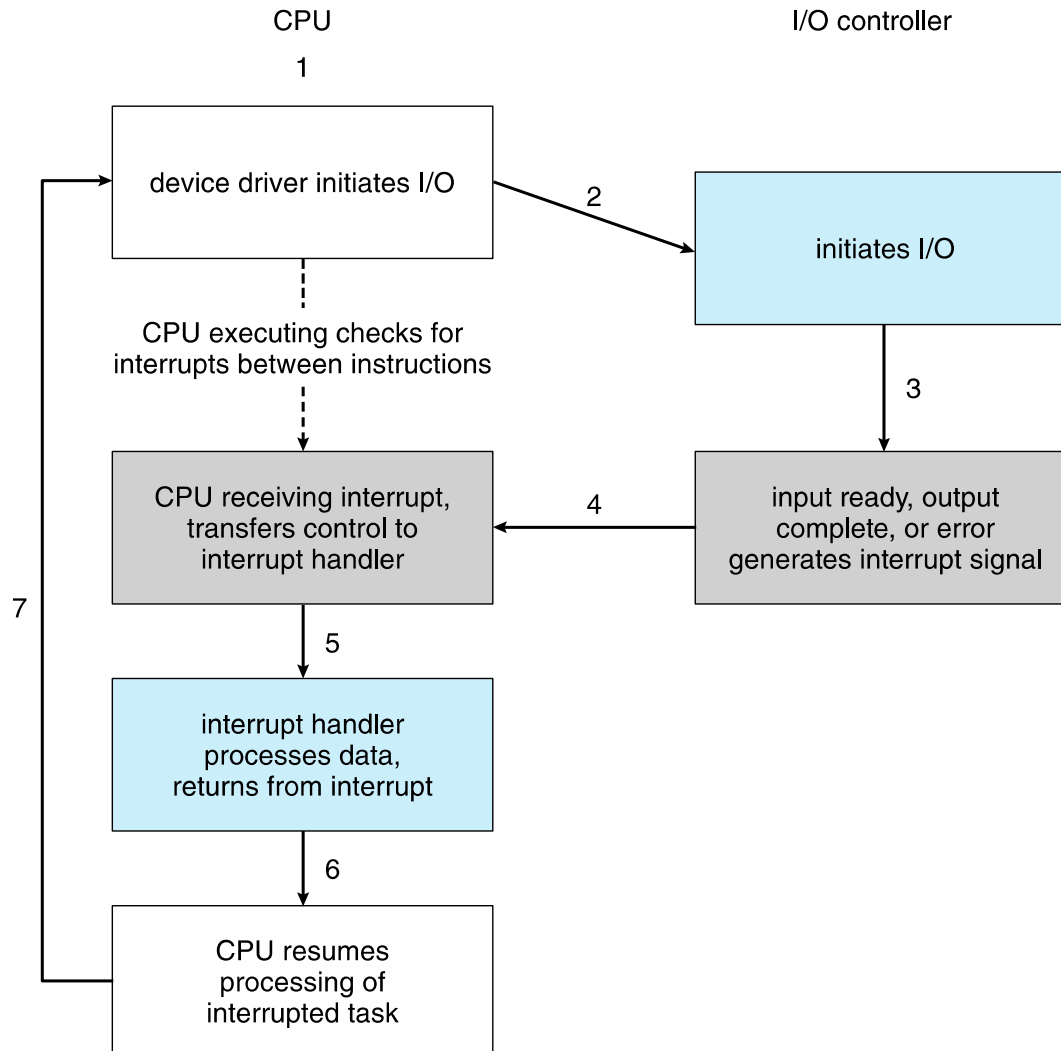
- Le système d'exploitation sauvegarde **l'état du processeur** en **stockant** les **registres** et le **compteur de programme**.
- Il identifie le type d'interruption survenu et exécute le code approprié pour chaque type d'interruption.
- Des segments de code distincts déterminent l'action à entreprendre pour chaque type d'interruption

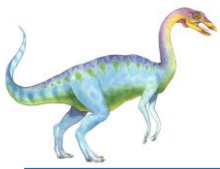




Interrupt-drive I/O Cycle

Cycle d'E/S du lecteur d'interruption





I/O Structure (Structure des E/S)

- Deux méthodes de gestion des E/S :
 - Le programme utilisateur **ne reprend** le contrôle **qu'une fois l'opération d'E/S terminée.**
 - Le programme utilisateur **récupère immédiatement** le contrôle après le lancement de l'opération d'E/S, **sans attendre sa finalisation.**





I/O Structure (Structure des E/S) (Cont.)

- Le contrôle revient au programme utilisateur **uniquement une fois les E/S terminées**
 - **L'instruction Wait** met le processeur en veille jusqu'à la prochaine interruption
 - **Boucle d'attente** (conflit d'accès à la mémoire)
 - **Au plus une demande d'E/S est en attente à la fois**, aucun traitement d'E/S simultané
- Le contrôle revient au programme utilisateur **sans attendre la fin des E/S**
 - **System call** (Appel système) - demande au système d'exploitation pour permettre à l'utilisateur d'attendre la fin des E/S
 - **Device-status table** (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 modifier l'entrée de la table pour inclure l'interruption





Programme de démarrage (Bootstrap)

- **Bootstrap program (chargement)** : Le programme de démarrage est exécuté au moment de la mise sous tension ou du redémarrage de l'ordinateur. (**power-up** or **reboot**)
- **Stockage** : Il est généralement stocké en mémoire **ROM** ou **EPROM**, ce qui signifie qu'il est intégré au matériel et ne peut pas être modifié facilement. On l'appelle aussi **firmware**.
- **Fonctionnement** :
 - Il **initialise** les **composants matériels** du système (processeur, mémoire, périphériques).
 - Il **charge le noyau** (**kernel**) système d'exploitation en mémoire.
 - Il **lance l'exécution** du **système d'exploitation** pour démarrer l'ordinateur,





Storage Structure

Structure de stockage

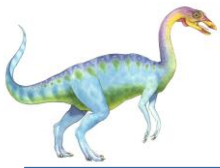




Structure de stockage

- **Mémoire principale** – seul support de stockage auquel le processeur peut accéder directement
 - Accès aléatoire **Random access**
 - Généralement **volatile**
 - Généralement mémoire à accès aléatoire sous la forme de mémoire vive dynamique (**DRAM**) **Dynamic Random-access Memory (DRAM)**
- **Stockage secondaire** – extension de la mémoire principale qui fournit une grande capacité de stockage **non volatile**





Structure de stockage

- **Hard Disk Drives (HDD)** – plateaux rigides en métal ou en verre recouverts d'un matériau d'enregistrement magnétique
 - La surface du disque est divisée logiquement en **pistes**, elles-mêmes subdivisées en **secteurs**
 - Le **contrôleur de disque** détermine l'interaction logique entre le périphérique et l'ordinateur
- **Non-volatile memory (NVM)**– plus rapides que les disques durs, non volatiles
 - Diverses technologies
 - De plus en plus populaires à mesure que la capacité et les performances augmentent, les prix baissent
 - ▶ **1. ROM (Read-Only Memory) – Mémoire en lecture seule** : PROM , EPROM , EEPROM
 - ▶ **2. Mémoire Flash (Type d'EEPROM plus rapide)** : SSD (Solid-State Drive) , Cartes mémoire (SD, microSD) , Clés USB
 - ▶ **3. NVRAM (Non-Volatile RAM)** : FRAM (Ferroelectric RAM) , MRAM (Magnetoresistive RAM) , PRAM (Phase-Change RAM) :





Storage Definitions and Notation Review

L'unité de base du stockage informatique est le **bit**, qui peut prendre deux valeurs : **0** ou **1**

Toutes les autres formes de stockage sont basées sur des regroupements de bits. Avec suffisamment de bits, un ordinateur peut représenter des chiffres, lettres, images, sons, vidéos, documents et programmes.

Octet (Byte) : Composé de **8 bits**, c'est l'unité de stockage la plus utilisée. La plupart des ordinateurs déplacent des octets plutôt que des bits.

Mot (Word) : Dépend de l'architecture du processeur.

Par exemple, **un processeur 64 bits** traite des **mots de 64 bits (8 octets)**.

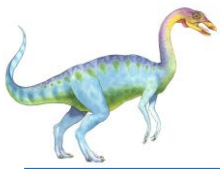
Unités de **mesure du stockage**. Le stockage et le débit des ordinateurs sont mesurés en octets et **multiples d'octets** :

1 KB (kilooctet)	=	1 024 octets
1 MB (mégaoctet)	=	1 024 ² octets
1 GB (gigaoctet)	=	1 024 ³ octets
1 TB (téraoctet)	=	1 024 ⁴ octets
1 PB (pétaoctet)	=	1 024 ⁵ octets

Cependant, les fabricants arrondissent souvent : 1 MB = 1 million d'octets,
et 1 GB = 1 milliard d'octets.

Exception : En réseau, la mesure se fait en **bits** car les données y sont transmises **bit** par **bit**

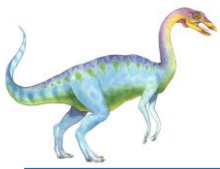




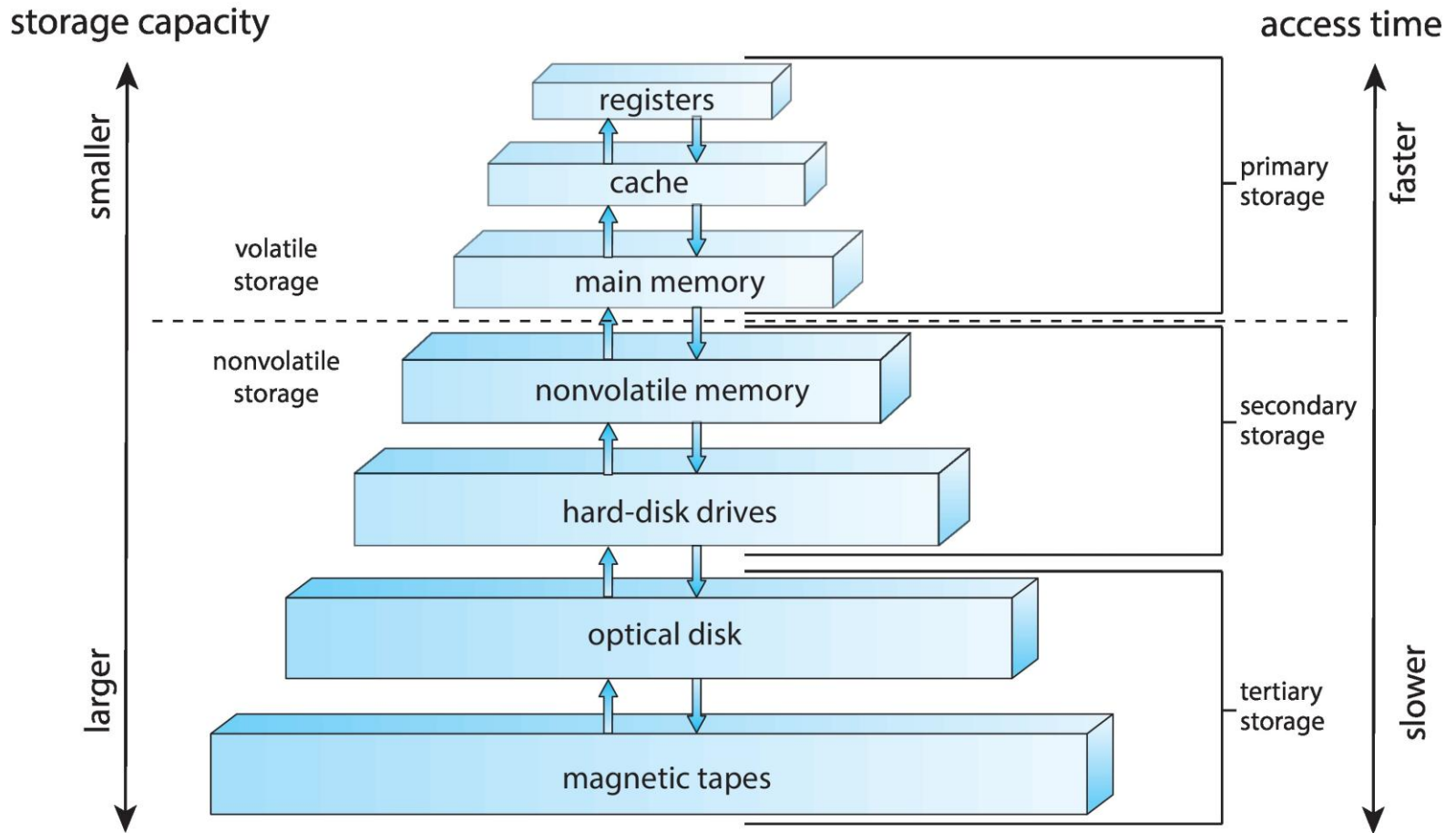
Hiérarchie de stockage

- Organisation hiérarchique des systèmes de stockage
 - Vitesse
 - Coût
 - Volatilité
- **Cache** – copie des informations dans un système de stockage plus rapide ; la mémoire principale peut être considérée comme un cache pour le stockage secondaire
- **Pilote de périphérique** pour chaque contrôleur de périphérique pour gérer les E/S
 - Fournit une interface uniforme entre le contrôleur et le noyau



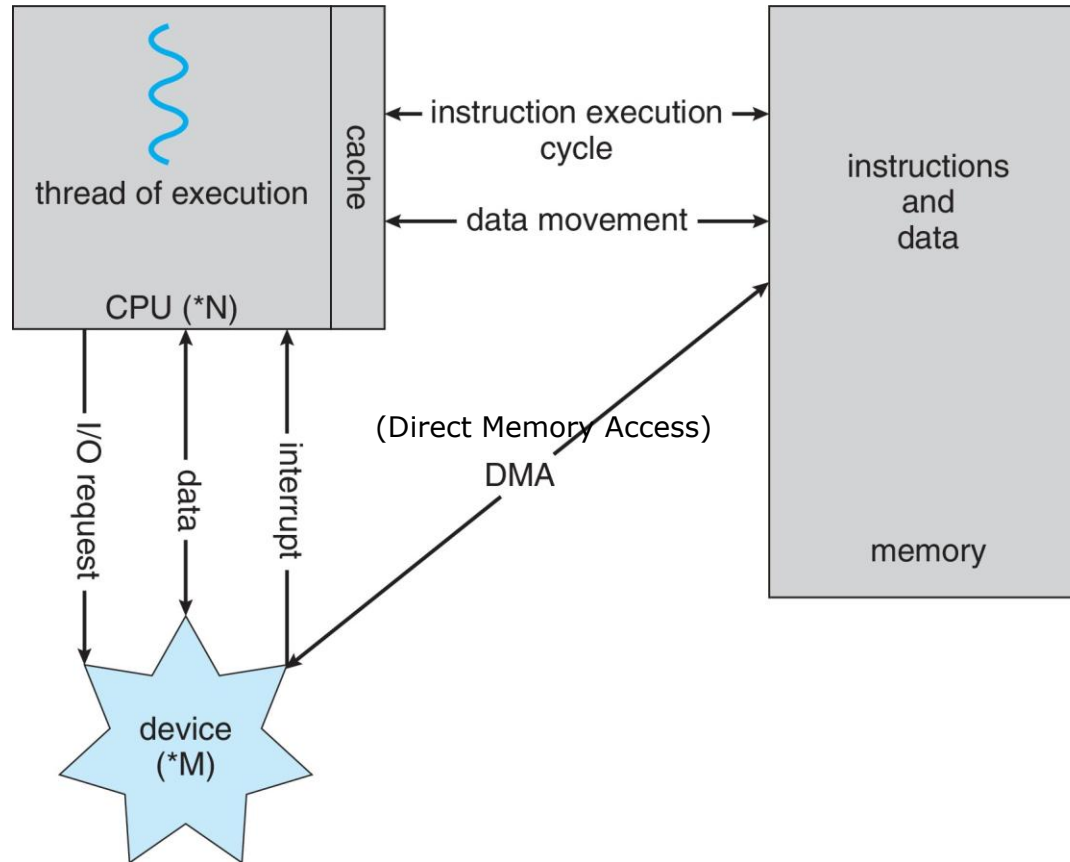


Hiérarchie des périphériques de stockage





Comment fonctionne un ordinateur moderne



A von Neumann architecture





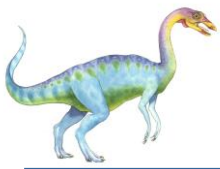
Opérations du système d'exploitation

1. **L'amorçage** : code simple pour initialiser le système, (charger le noyau)
2. **Chargement du noyau**
3. **Démarrage des démons système** (services fournis en dehors du noyau)
4. **Interruption du noyau pilotée** (matérielle et logicielle)
 - Interruption matérielle par l'un des périphériques
 - Interruption logicielle (**exception** ou **trap**) :
 - ▶ Erreur d'application (par exemple, division par zéro)
 - ▶ Demande de service du système d'exploitation – **appel système**
 - ▶ D'autres problèmes de processus incluent une boucle infinie, des processus se modifiant les uns les autres ou le système d'exploitation

Exceptions = erreurs imprévues ou conditions anormales.

Traps = interruptions volontaires pour demander des services au SE.

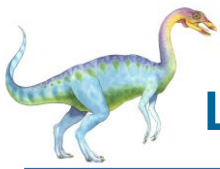




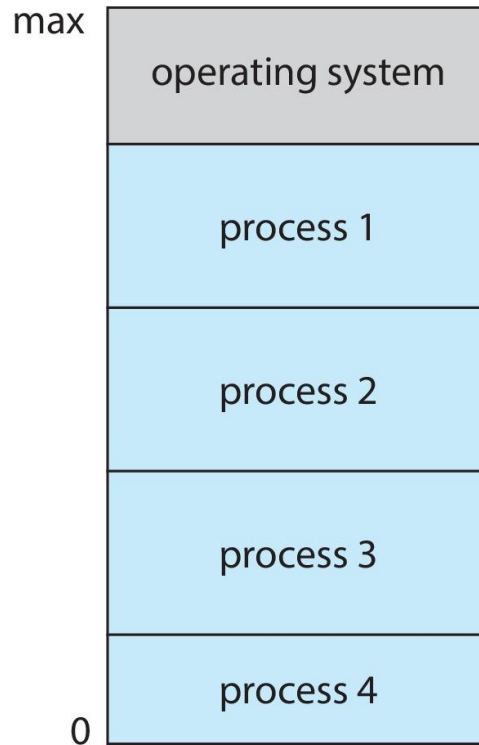
Multiprogramming (Batch system: Système par lots)

- Un **utilisateur unique ne peut pas toujours occuper le processeur** et les périphériques d'E/S
- La **multiprogrammation** organise les tâches de sorte que **le processeur en ait toujours une tâche à exécuter**
- Un **moment donné, seules certaines tâches** sont gardées en mémoire et **une seule tâche sélectionnée et exécutée** via la planification des tâches (**job scheduling**)
- **Lorsque la tâche doit attendre** (pour les E/S par exemple), le **SE passe à une autre tâche**





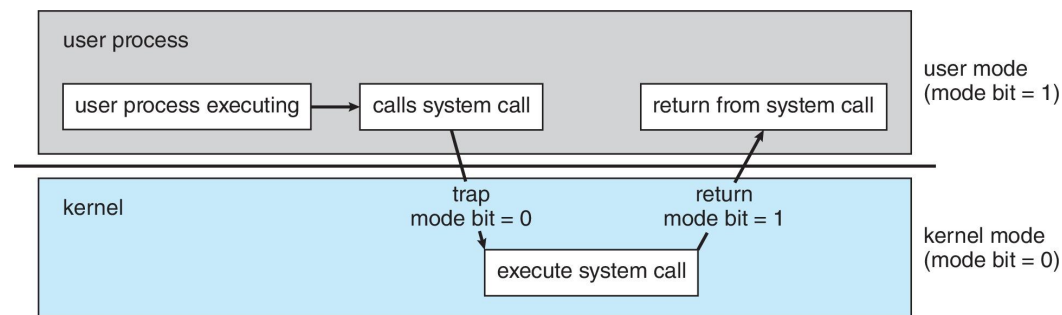
La disposition de la mémoire pour un système multiprogrammé

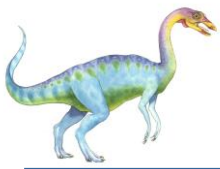




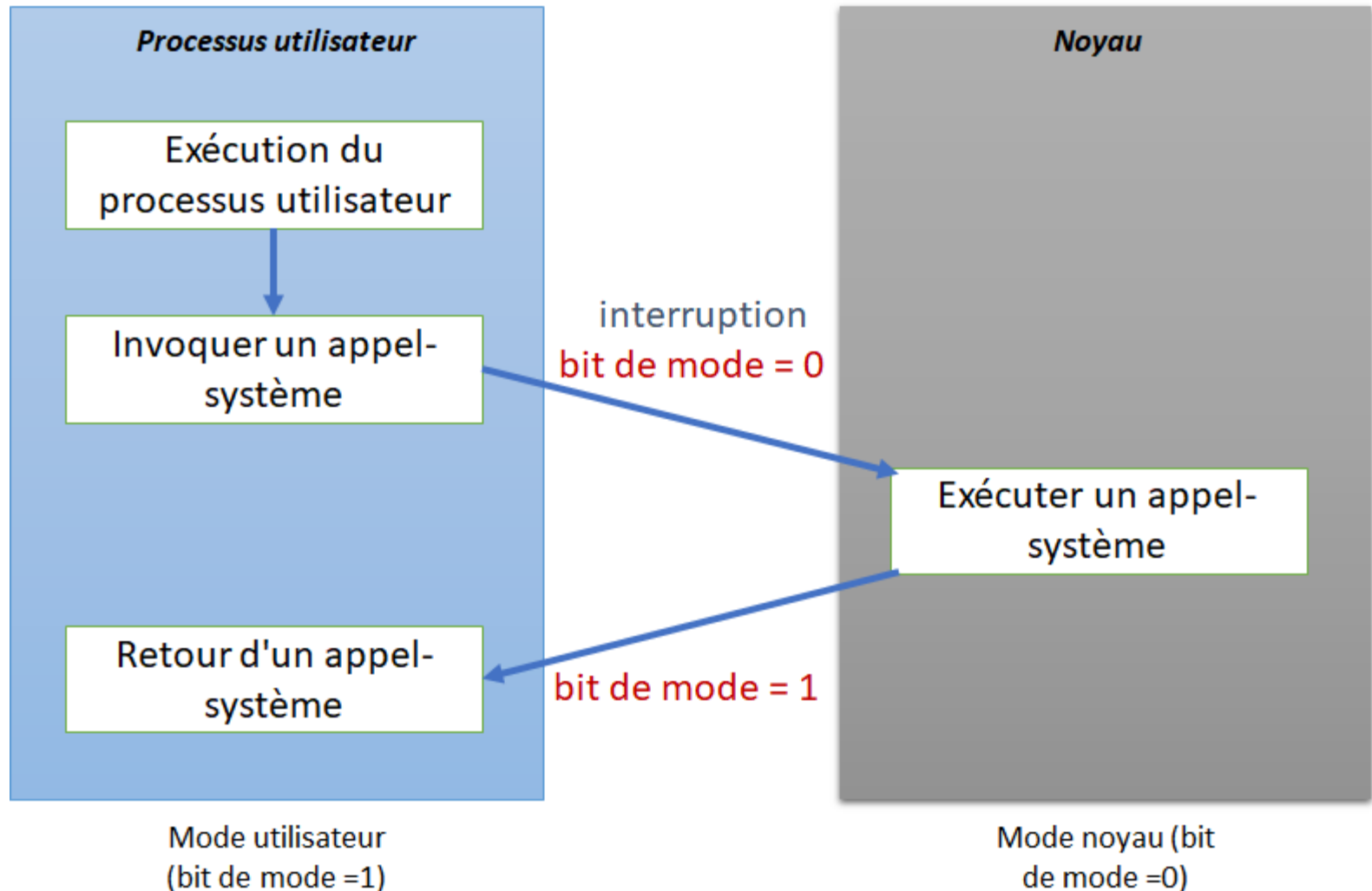
Fonctionnement en mode double

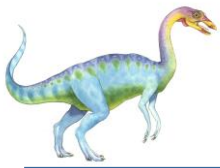
- L'opération en **mode double** permet au système d'exploitation de se protéger lui-même ainsi que les autres composants du système.
- **User mode (mode utilisateur)** et **mode Kernel (mode noyau)**
- **Mode bit** fourni par le matériel
 - Il permet de distinguer quand le système exécute du code utilisateur ou du code noyau.
 - Lorsque l'utilisateur est en exécution → le bit de mode bit est « **user** »
 - Lorsque le code noyau est en cours d'exécution → le mode bit est « **Kernel** »
- Comment garantir que l'utilisateur ne définisse pas explicitement le bit de mode sur « noyau » ?
 - Un appel système change le mode en noyau, et un retour de l'appel le réinitialise en mode utilisateur.
- Certaines instructions sont désignées comme **privilégiées** et ne peuvent être exécutées qu'en mode noyau. .





Transition du mode utilisateur au mode noyau

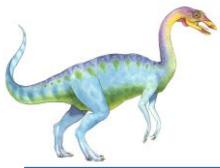




Timer (minuterie)

- Timer : La **minuterie** pour éviter les boucles infinies (ou l'occupation excessive des ressources par un processus)
 - La **minuterie est configurée pour interrompre** l'ordinateur après un certain délai.
 - **Un compteur est maintenu et est décrémenté par l'horloge physique.**
 - Le système d'exploitation configure le compteur (**instruction privilégiée**).
 - **Lorsque le compteur atteint zéro, une interruption est générée.**
 - La minuterie **est mise en place par (job scheduler)** pour contrôler le temps alloué a chaque processus.





Gestion des processus

- Un processus est un programme en exécution. C'est une unité de travail au sein du système. Un programme est une **entité passive**, tandis qu'un processus est **une entité active**.
 - Le **processus nécessite des ressources** pour accomplir sa tâche :
 - ▶ CPU, memory, I/O, files
 - ▶ Données d'initialisation
 - La **terminaison** d'un processus **nécessite la récupération de toutes les ressources** réutilisables.
 - Un processus à **un seul fil d'exécution (monothreadé)** possède un seul **compteur de programme** qui spécifie l'emplacement de la prochaine instruction à exécuter.
 - Un processus **multithreadé** possède **un compteur de programme par fil d'exécution**.
 - Le processus exécute les instructions de manière séquentielle, une à la fois, jusqu'à ce qu'il soit terminé.
- En général, un système a **de nombreux processus**, certains appartenant à **l'utilisateur**, d'autres au **système d'exploitation**, qui s'exécutent simultanément sur un ou plusieurs processeurs.
- La concurrence est obtenue en multipliant l'utilisation des processeurs entre les processus / fils d'exécution.





Activités de gestion des processus

- Dans le cadre de la gestion des processus, le système d'exploitation est responsable des activités suivantes :
 - **Création** et **suppression** de *processus* utilisateur et système
 - **Suspendre** et **reprendre** des *processus*
 - Fournir des mécanismes pour la **synchronisation** des *processus*
 - Fournir des mécanismes pour la **communication** des *processus*
 - Fournir des mécanismes pour la **gestion des blocages**

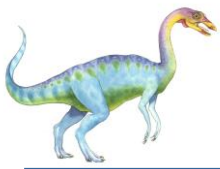




Gestion de la mémoire

- Pour exécuter un programme, **toutes (ou une partie) des instructions doivent être en mémoire**
- Toutes (ou une partie) des données nécessaires au programme doivent être en mémoire
- La gestion de la mémoire détermine ce qui se trouve en mémoire et quand
 - Optimisation de l'utilisation du processeur et de la réponse de l'ordinateur aux utilisateurs
- Activités de gestion de la mémoire
 - Suivi des parties de la mémoire actuellement utilisées et par qui
 - Décision des processus (ou parties de ceux-ci) et des données à déplacer vers et hors de la mémoire
 - Allocation et dé-allocation de l'espace mémoire selon les besoins





Gestion du système de fichiers

- Le système d'exploitation fournit une vue logique et uniforme du stockage des informations
 - L'abstraction des propriétés physiques de l'unité de stockage logique – **fichier**
 - Chaque support est contrôlé par un périphérique (par exemple, lecteur de disque, lecteur de bande)
 - ▶ Les propriétés variables incluent la vitesse d'accès, la capacité, le taux de transfert de données, la méthode d'accès (séquentielle ou aléatoire)
- **Gestion du système de fichiers**
 - **Fichiers généralement organisés en répertoires**
 - **Contrôle d'accès** sur la plupart des systèmes pour déterminer **qui peut accéder à quoi**
 - Les activités du SE comprennent
 - ▶ **Création** et **suppression** de fichiers et de répertoires
 - ▶ **Primitives** pour manipuler les fichiers et les répertoires
 - ▶ **Mappage** de fichiers sur un stockage secondaire
 - ▶ **Sauvegarde** de fichiers sur un support de stockage stable (non volatil)

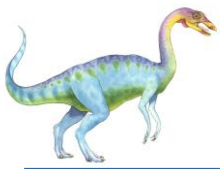




Gestion du stockage de masse

- Généralement, les disques sont utilisés pour stocker des données qui ne tiennent pas dans la mémoire principale ou des données qui doivent être conservées pendant une « longue » période.
- Une gestion appropriée est d'une importance capitale.
- La vitesse de fonctionnement de l'ordinateur dépend entièrement du sous-système de disque et de ses algorithmes.
- Activités du système d'exploitation.
 - Montage et démontage.
 - Gestion de l'espace libre.
 - Allocation de stockage.
 - Planification des disques.
 - Partitionnement.
 - Protection

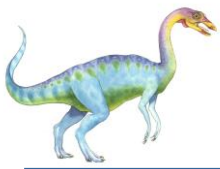




Mise en cache

- Principe clé, appliqué à plusieurs niveaux dans un ordinateur (matériel, système d'exploitation, application)
- Les informations **en cours d'utilisation** sont **temporairement copiées** d'un stockage plus lent vers un **stockage plus rapide**.
- Le stockage rapide (**cache**) est d'abord vérifié 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 depuis cet emplacement.
- Le cache est plus petit que le stockage principal.
 - Un problème de conception majeur dans la gestion du cache est la **taille du cache** et la **politique de remplacement**.





Caractéristiques des différents types de stockage

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

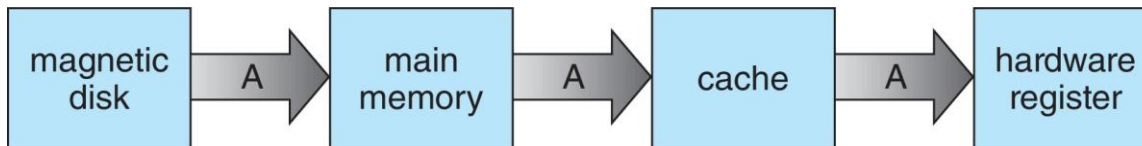
Le mouvement entre les niveaux de hiérarchie de stockage peut être **explicite** ou **implicite**





Migration des données 'A' du disque vers le registre

- Les environnements multitâches doivent veiller à utiliser la valeur la plus récente, quel que soit l'endroit où elle est stockée dans la hiérarchie de stockage.



- Les environnements multiprocesseurs doivent garantir la cohérence des caches au niveau matériel, de sorte que toutes les unités centrales (CPU) possèdent la valeur la plus récente dans leur cache.
- La situation est encore plus complexe dans un environnement distribué.
 - Plusieurs copies d'une donnée peuvent exister.
 - Diverses solutions sont abordées dans le Chapitre 19: Réseaux et systèmes distribués





Sous-système d'E/S (Entrée/Sortie)

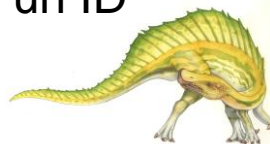
- L'un des objectifs du système d'exploitation est de cacher les particularités des périphériques matériels à l'utilisateur
- **Sous-système d'E/S responsable de**
 - Gestion de la mémoire des E/S, y compris la mise en mémoire tampon (stockage temporaire des données pendant leur transfert), la mise en cache (stockage de parties de données dans un stockage plus rapide pour des performances accrues), la mise en file d'attente (chevauchement de la sortie d'une tâche avec l'entrée d'autres tâches)
 - Interface générale du pilote de périphérique
 - Pilotes pour des périphériques matériels spécifiques





Protection et sécurité

- **Protection** – tout mécanisme de contrôle de 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
 - Large gamme, y compris le déni de service, les vers, les virus, le vol d'identité, le vol de service
- Les systèmes font généralement d'abord la distinction entre les utilisateurs, pour déterminer qui peut faire quoi
 - Les identités des utilisateurs (**user IDs**, security IDs) incluent le nom et le numéro associé, un par utilisateur
 - L'ID utilisateur est ensuite associé à tous les fichiers et processus de cet utilisateur pour déterminer le contrôle d'accès
 - L'identifiant de groupe (**group ID**) permet de définir un ensemble d'utilisateurs et de gérer les contrôles, puis de l'associer également à chaque processus, fichier
 - **L'escalade des privilèges** permet à l'utilisateur de passer à un ID effectif avec plus de droits

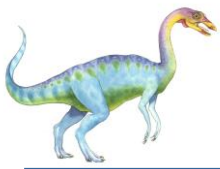




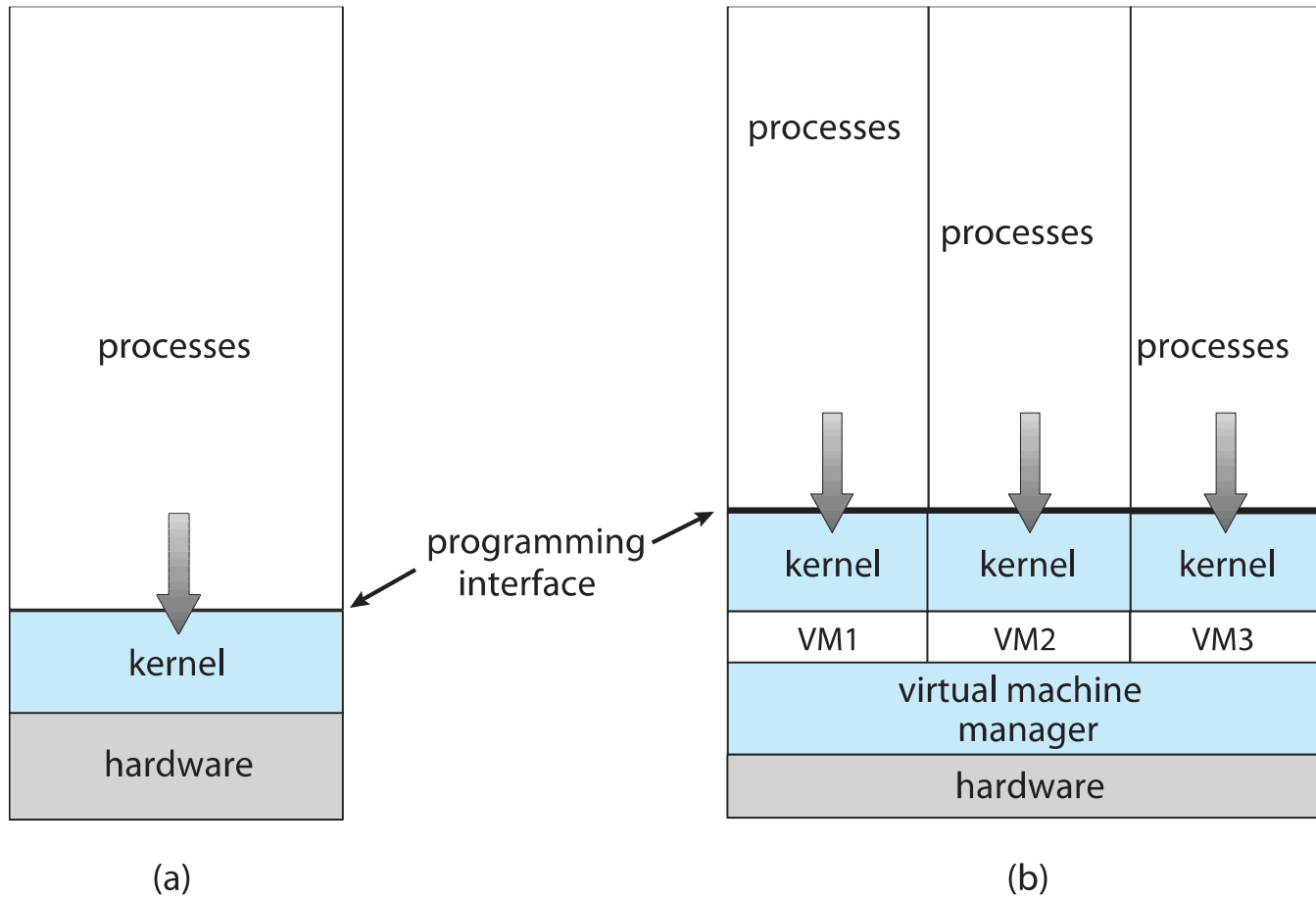
Virtualisation

- Permet aux systèmes d'exploitation d'exécuter des applications dans d'autres OS.
 - Industrie en pleine expansion.
- **Émulation** : utilisée lorsque le processeur source est différent du processeur cible (ex. PowerPC vers Intel x86).
 - C'est généralement la méthode la plus lente.
 - **Interprétation** : utilisée lorsque le langage informatique n'est pas compilé en code natif, donc exécuté ligne par ligne.
- **Virtualisation** : l'OS invité et l'OS hôte sont compilés pour le même processeur.
 - **Virtualisation complète** : L'hyperviseur émule entièrement le matériel, permettant d'exécuter un système d'exploitation invité sans modification. **Ex:** **VMware** er, **Microsoft Hyper-V** , **VirtualBox**
 - **Para-virtualisation** : L'OS invité est modifié pour interagir directement avec l'hyperviseur , **Ex.** **Citrix XenServer**
 - **Virtualisation au niveau du système d'exploitation** : Plusieurs instances isolées (conteneurs) partagent le même noyau du système d'exploitation hôte. **Ex :** **Docker**, **LXC**, **OpenVZ**
 - **Virtualisation matérielle assistée** : Utilise des fonctionnalités spécifiques du processeur (Intel VT-x, AMD-V) pour améliorer l'exécution des machines virtuelles. **Ex :** **KVM (Kernel-based Virtual Machine)**, **VMware ESXi**





Virtualisation

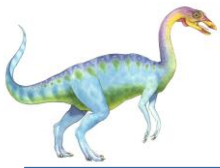




Systèmes distribués

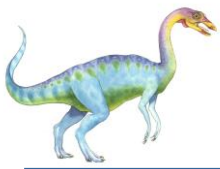
- Ensemble de systèmes distincts, parfois hétérogènes, connectés en réseau.
 - Le réseau est un chemin de communication, le plus courant étant **TCP/IP**.
- **Types de réseaux :**
 - **LAN (Local Area Network)** : réseau local (ex. entreprise, maison).
 - **WAN (Wide Area Network)** : réseau étendu (ex. Internet).
 - **MAN (Metropolitan Area Network)** : réseau métropolitain (ex. ville).
 - **PAN (Personal Area Network)** : réseau personnel (ex. Bluetooth, Wi-Fi entre appareils).
- Le **système d'exploitation réseau (Network OS)** permet l'échange d'informations entre les systèmes.
- Un **schéma de communication** facilite l'envoi et la réception de messages entre machines.
- L'objectif est de donner l'**illusion d'un système unique** malgré la distribution des machines.





Computer System Architecture

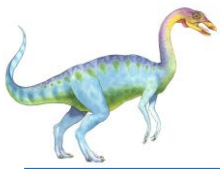




Architecture des systèmes informatiques

- **La plupart des systèmes utilisent un seul processeur généraliste.**
 - Cependant, ils **intègrent** aussi des **processeurs spécialisés** (Ex. GPU).
- Les **systèmes multiprocesseurs** sont de plus en plus courants et importants.
 - Également appelés **systèmes parallèles** ou **systèmes fortement couplés**.
 - **Avantages :**
 1. **Meilleur rendement** (plus de tâches exécutées simultanément).
 2. **Réduction des coûts** (partage des ressources).
 3. **Fiabilité accrue** (dégradation progressive ou tolérance aux pannes).
 - **Types de multiprocesseurs :**
 1. **1. Multiprocessing asymétrique** : chaque processeur a une tâche spécifique.
 2. **2. Multiprocessing symétrique** : tous les processeurs exécutent les mêmes tâches.

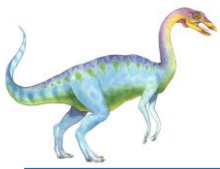




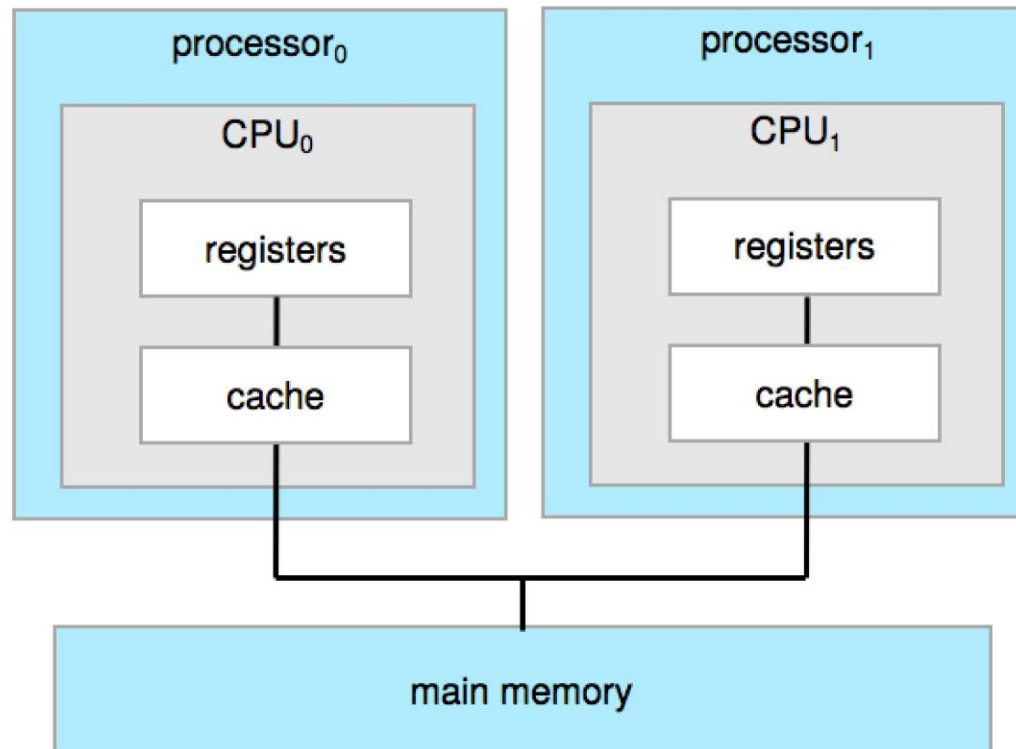
Architecture des systèmes informatiques

- Most systems use a single general-purpose processor
 - Most systems have special-purpose processors as well
- **Multiprocessors** systems growing in use and importance
 - Also known as **parallel systems**, **tightly-coupled systems**
 - Advantages include:
 1. **Increased throughput**
 2. **Economy of scale**
 3. **Increased reliability** – graceful degradation or fault tolerance
 - Two types:
 1. **Asymmetric Multiprocessing** – each processor is assigned a specific task.
 2. **Symmetric Multiprocessing** – each processor performs all tasks





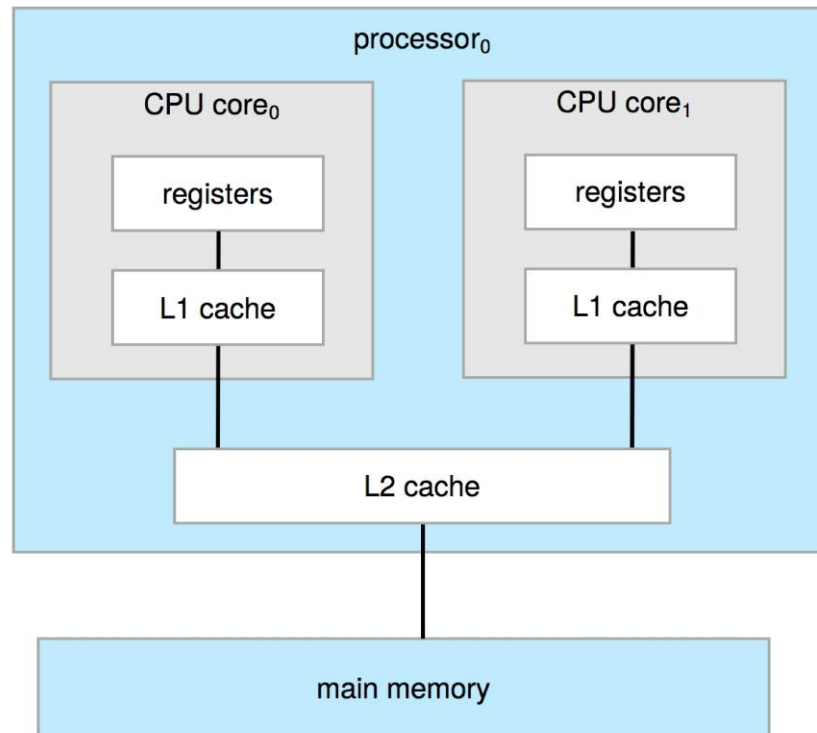
Architecture multiprocesseur symétrique





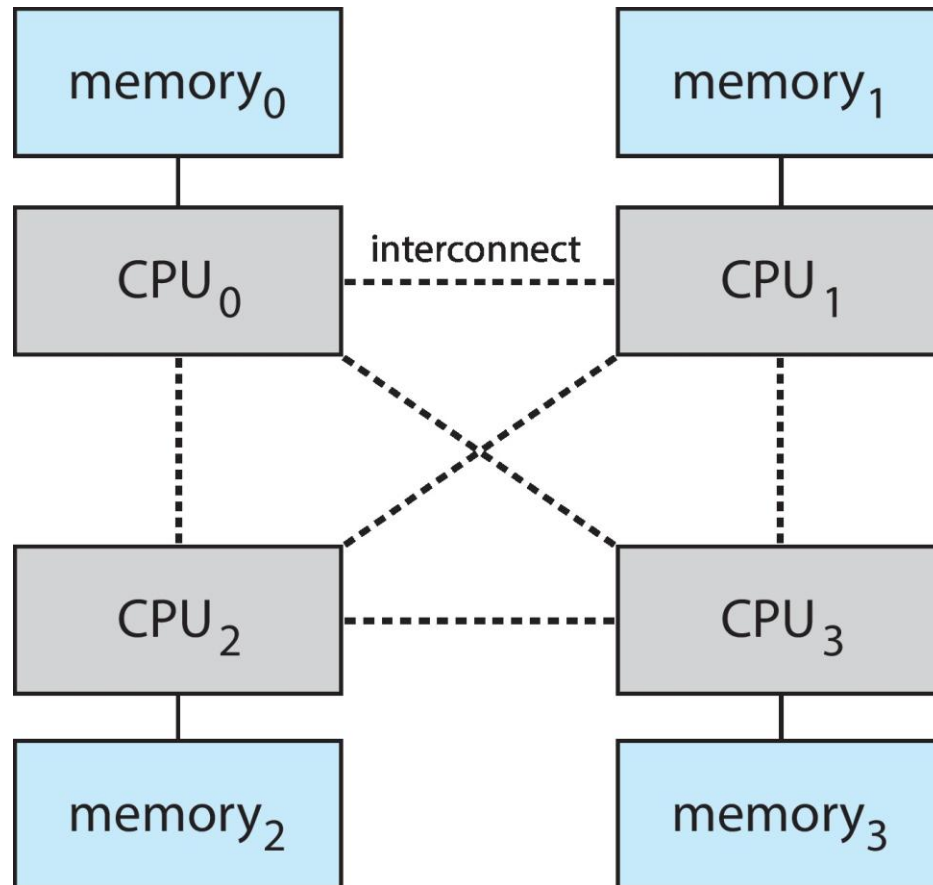
Conception à double cœur (Dual-Core)

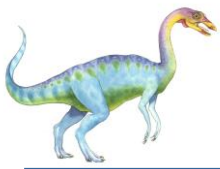
- Systèmes multipuces et **multicœurs**
- Systèmes contenant toutes les puces
 - Châssis contenant plusieurs systèmes distincts





Système d'accès à la mémoire non uniforme





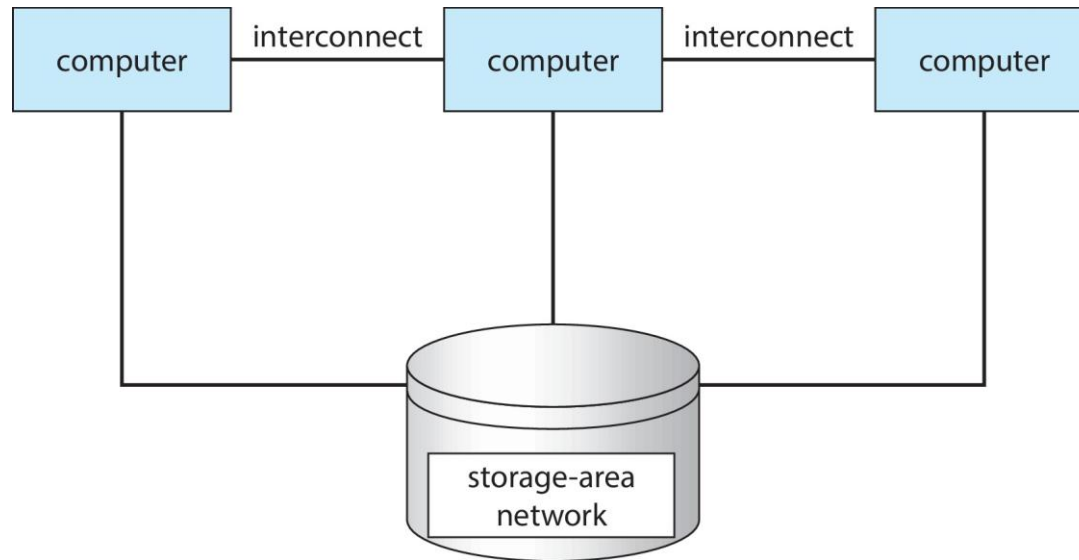
Systèmes groupés “Clustered”

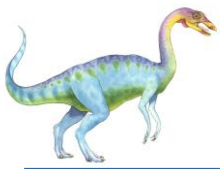
- Similaire aux systèmes multiprocesseurs, mais avec plusieurs systèmes travaillant ensemble.
 - Généralement partagent un **stockage commun** via un **SAN (Storage Area Network)**.
 - Offre un **service à haute disponibilité (HA)**, qui continue de fonctionner en cas de panne.
- **Types de clustering :**
 - **Clustering asymétrique** : une machine est en veille (hot-standby) et prend le relais en cas de défaillance.
 - **Clustering symétrique** : plusieurs nœuds exécutent des applications et se surveillent mutuellement.
- Certains clusters sont dédiés au **calcul haute performance (HPC)**.
 - Les applications doivent être conçues pour **exploiter le parallélisme**.
- Certains clusters utilisent un **gestionnaire de verrouillage distribué (DLM)** pour éviter les conflits d'accès.





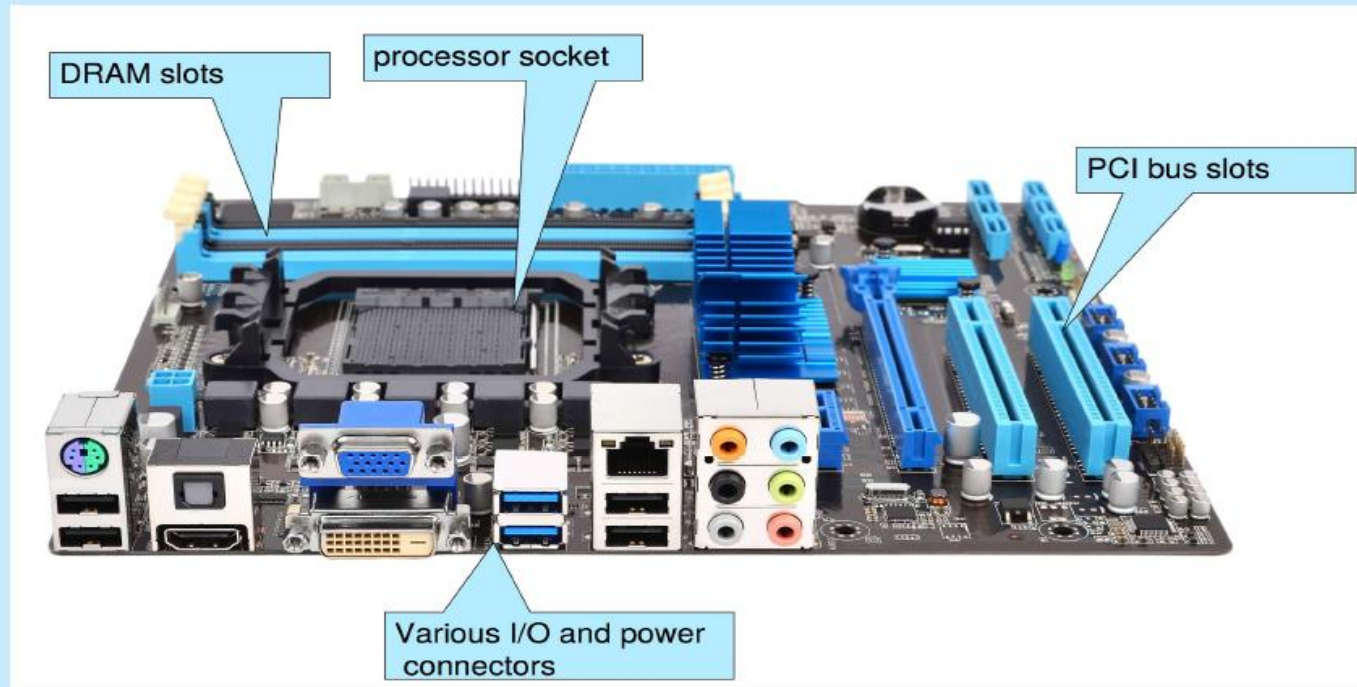
Clustered Systems





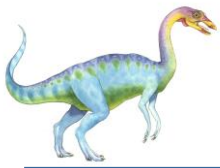
PC Motherboard

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



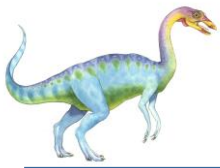
This board is a fully-functioning computer, once its slots are populated. It consists of a processor socket containing a CPU, DRAM sockets, PCIe bus slots, and I/O connectors of various types. Even the lowest-cost general-purpose CPU contains multiple cores. Some motherboards contain multiple processor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





Computer System Environments

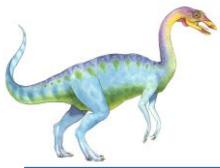




Computing Environments

- Traditionnel
- Mobile
- Client Serveur
- Peer-to-Peer
- Cloud computing
- Temps réel Embarqué (Real-time Embedded)

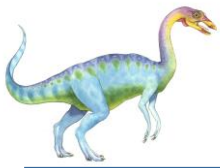




Traditional

- **Machines autonomes**, mais souvent interconnectées (ex. Internet).
- Les portails web donnent accès aux systèmes internes.
- Les **clients légers** fonctionnent comme des terminaux web.
- Les appareils **mobiles** utilisent des **réseaux sans fil**.
- Les réseaux sont omniprésents, même à domicile avec des **pare-feu** pour la sécurité.





Mobile

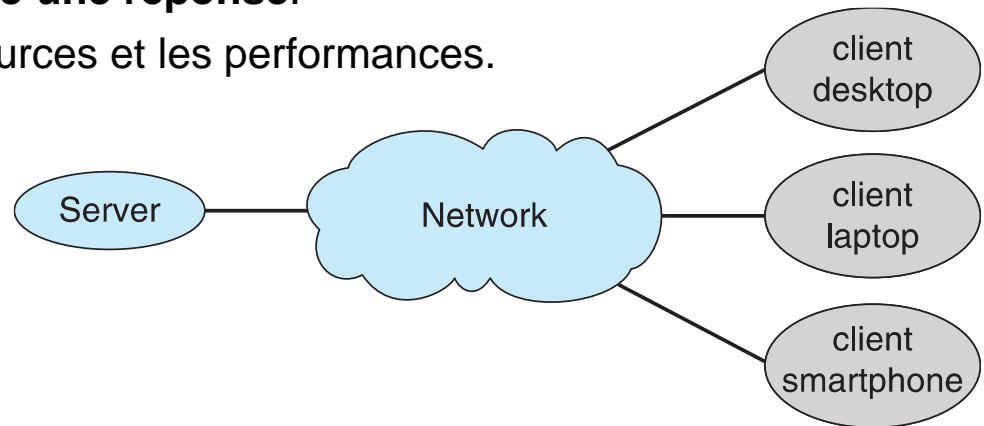
- **Smartphones, tablettes**, etc.
- Différents des **PC portables** par leurs fonctionnalités supplémentaires (GPS, gyroscope).
- Permettent des applications innovantes (réalité augmentée).
- Utilisez les réseaux de données **sans fil (IEEE 802.11)** ou **cellulaires** pour la connectivité
- Principaux systèmes : **iOS (Apple)** et **Android (Google)**.





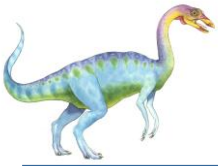
Client Server

- L'architecture **client-serveur** repose sur la répartition des tâches entre des clients, qui font des demandes, et des serveurs, qui y répondent.
 - Le **client envoie** une **requête** via un réseau.
 - Le **serveur la traite** et **renvoie une réponse**.
 - Ce modèle optimise les ressources et les performances.



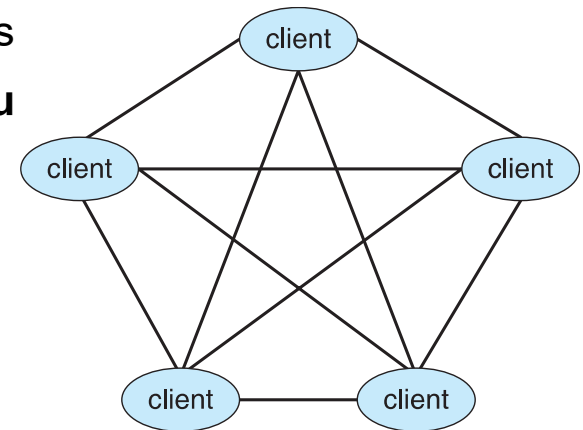
- **Types de serveurs :**
 - **Calcul** : exécute des traitements complexes (ex. **bases de données**).
 - **Fichiers** : gère le stockage et le **partage de fichiers**.
 - **Web** : **héberge** sites et applications.
 - **Applications** : exécute des logiciels métiers (ex. **ERP, CRM**).
- **Largement utilisé** en entreprise et dans le cloud, il assure centralisation, évolutivité et efficacité.





Peer-to-Peer

- Un autre modèle de **système distribué**
- Le **P2P ne fait pas de distinction** entre les **clients** et les **serveurs**
 - 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 la demande de service et répond aux demandes de service via le **protocole de découverte**
 - Les exemples incluent **Napster** et **Gnutella**, la voix sur IP (**VoIP**) comme **Skype**

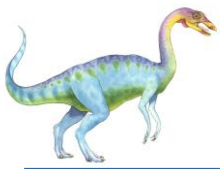




Cloud Computing

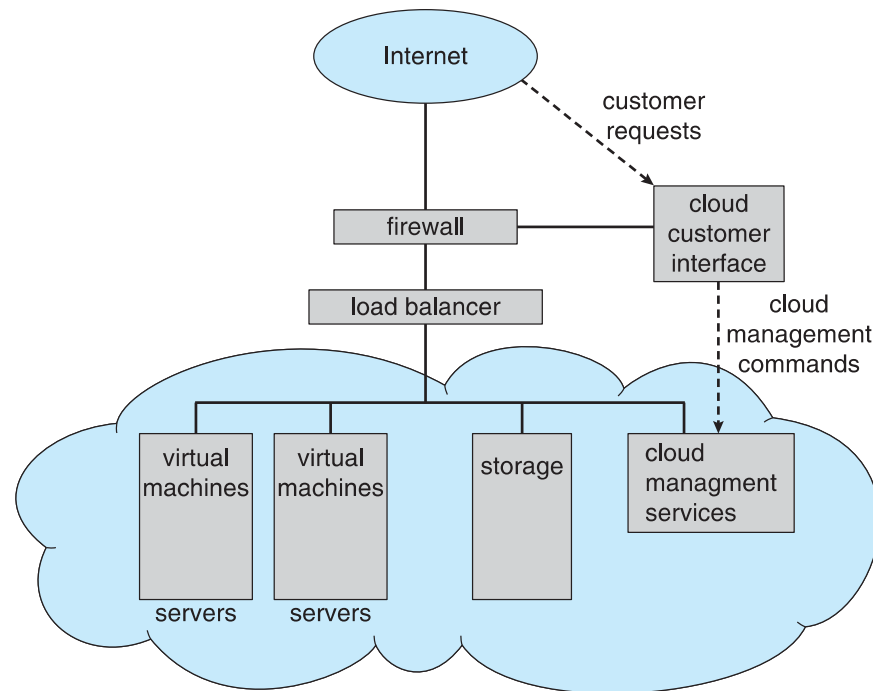
- **Fournit des services** de **calcul**, de **stockage** et même des **applications en tant que service** sur un réseau
- **Extension logique de la virtualisation** car elle utilise la virtualisation comme base de ses fonctionnalités.
- **Types**
 - **Cloud public** – disponible via Internet pour toute personne prête à payer
 - **Cloud privé** – géré par une entreprise pour son propre usage
 - **Cloud hybride** – comprend 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 Office365 , Google Doc, etc)
 - **Plateforme en tant que service (PaaS)** – pile logicielle prête à être utilisée par l'application via Internet (par exemple, **Google Colab Kaggle** , serveur de base de données)
 - **Infrastructure en tant que service (IaaS)** – serveurs ou stockage disponibles sur Internet (par exemple, stockage disponible pour une utilisation de sauvegarde)

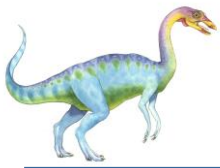




Cloud Computing (cont.)

- Environnements de cloud computing composés de systèmes d'exploitation traditionnels, de VMM et d'outils de gestion du cloud
 - La connectivité Internet nécessite des mesures de sécurité telles que des pare-feu
 - Les équilibreurs de charge répartissent le trafic sur plusieurs applications

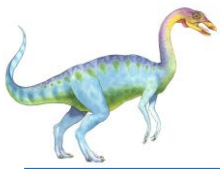




Real-Time Embedded Systems

- Real-time embedded systems most prevalent form of computers
 - Vary considerable, special purpose, limited purpose OS, **real-time OS**
 - Use expanding
- Many other special computing environments as well
 - Some have OSes, some perform tasks without an OS
- Real-time OS has well-defined fixed time constraints
 - Processing ***must*** be done within constraint
 - Correct operation only if constraints met

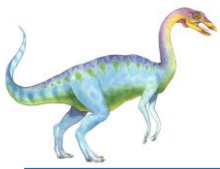




Real-Time Embedded Systems

- Les **systèmes embarqués temps réel** sont très répandus. Ils varient en fonction de leur usage, avec des systèmes d'exploitation dédiés ou sans OS.
- Les **systèmes temps réel** ont des contraintes de **temps strictes**. Les **traitements** doivent être **effectués** dans ces **délais** pour garantir un fonctionnement correct.
- Ils sont utilisés dans des domaines nécessitant des **réactions rapides** comme **l'automobile**, **l'aérospatial**, et le **contrôle industriel**.





Systèmes d'exploitation libres et open source

- Systèmes d'exploitation **mis à disposition sous forme de code source** plutôt que simplement binaires, fermés et propriétaires
- Mouvement de protection contre la copie et de gestion des droits numériques **Digital Rights Management (DRM)**
- Lancé par la **Free Software Foundation (FSF)**, qui a la licence publique **GNU Public License (GPL)**
 - **Le logiciel libre** et le **logiciel open source** sont **deux idées différentes** défendues par différents groupes de personnes
 - ▶ <https://www.gnu.org/philosophy/open-source-misses-the-point.en.html>
- Les exemples incluent GNU/Linux et BSD UNIX (y compris le cœur de Mac OS X), et bien d'autres
- Peut utiliser VMM comme VMware Player (gratuit sur Windows), Virtualbox (open source et gratuit sur de nombreuses plateformes - <http://www.virtualbox.com>)
 - Utiliser pour exécuter des systèmes d'exploitation invités pour l'exploration





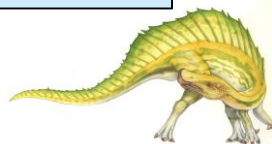
The Study of Operating Systems

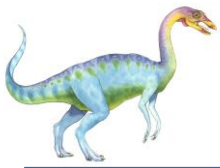
There has never been a more interesting time to study operating systems, and it has never been easier. The open-source movement has overtaken operating systems, causing many of them to be made available in both source and binary (executable) format. The list of operating systems available in both formats includes Linux, BSD UNIX, Solaris, and part of macOS. The availability of source code allows us to study operating systems from the inside out. Questions that we could once answer only by looking at documentation or the behavior of an operating system we can now answer by examining the code itself.

Operating systems that are no longer commercially viable have been open-sourced as well, enabling us to study how systems operated in a time of fewer CPU, memory, and storage resources. An extensive but incomplete list of open-source operating-system projects is available from https://curlie.org/Computers/Software/Operating_Systems/Open_Source/

In addition, the rise of virtualization as a mainstream (and frequently free) computer function makes it possible to run many operating systems on top of one core system. For example, VMware (<http://www.vmware.com>) provides a free “player” for Windows on which hundreds of free “virtual appliances” can run. Virtualbox (<http://www.virtualbox.com>) provides a free, open-source virtual machine manager on many operating systems. Using such tools, students can try out hundreds of operating systems without dedicated hardware.

The advent of open-source operating systems has also made it easier to make the move from student to operating-system developer. With some knowledge, some effort, and an Internet connection, a student can even create a new operating-system distribution. Just a few years ago, it was difficult or impossible to get access to source code. Now, such access is limited only by how much interest, time, and disk space a student has.

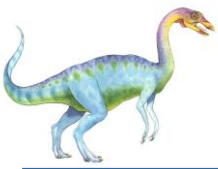




Kernel Data Structure

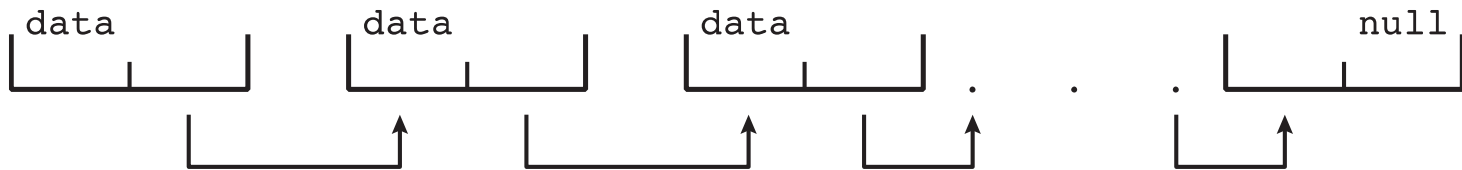
Structure des données du noyau



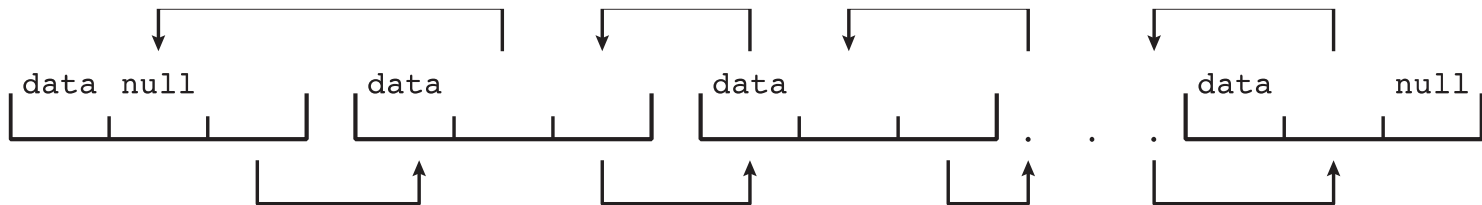


Structure des données du noyau

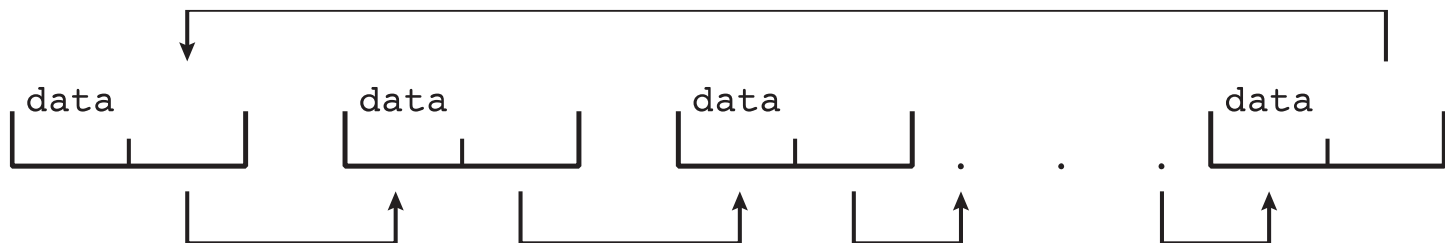
- De nombreuses structures de données de programmation similaires aux standards
- **Liste chaînée simple**



- **Liste doublement chaînée**



- **Liste chaînée circulaire**



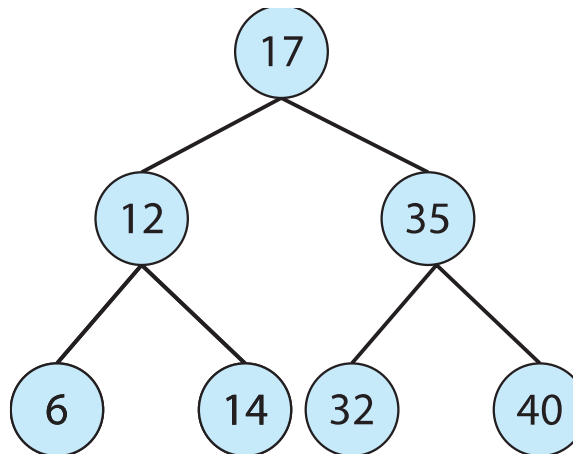


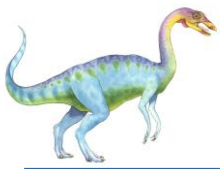
Structure des données du noyau

■ Arbre de recherche binaire

gauche \leq droite

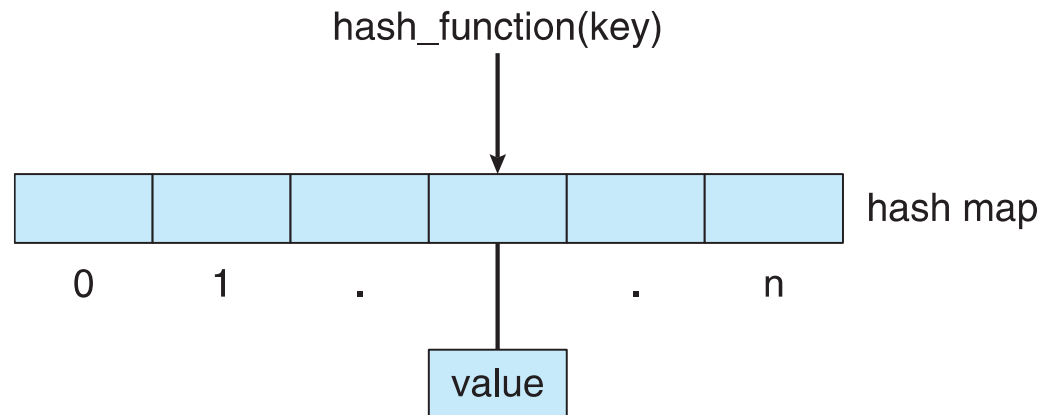
- Les performances de recherche sont $O(n)$
- **Arbre de recherche binaire équilibré** $O(\lg n)$





Kernel Data Structures

- La fonction de hachage peut créer une **carte de hachage (hash map)**



- **Bitmap** – Un bitmap est une suite de bits (0 ou 1) utilisée pour indiquer l'état de plusieurs éléments, par exemple pour marquer des ressources utilisées ou libres.
- Linux utilise diverses structures de données pour organiser et gérer efficacement l'information dans le noyau :
 - Listes chaînées pour une gestion flexible des éléments, `<linux/list.h>`
 - Files FIFO pour traiter les données dans l'ordre d'arrivée, `<linux/kfifo.h>`
 - Arbres rouge-noir pour une recherche rapide et équilibrée. `<linux/rbtree.h>`



End of Chapter 1

