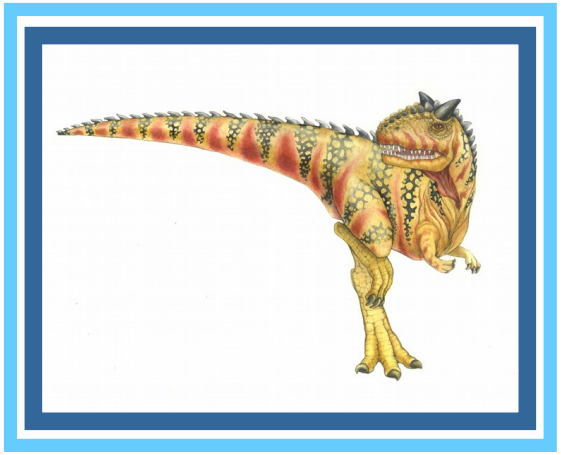
Chapitre 1: Introduction



Chapitre 1: Introduction

- Ce que fait un système d'exploitation
- Système informatique
  - Organisation
  - Architecture
  - Structure
  - Opérations
- Gestion
  - Processus
  - Memoire
  - Stockage
- Protection et sécurité
- Structures de données du noyaux
- Environnements informatiques

DIRO

Objectifs

- Décrire l'ogranisation de base d'un système informatique
- Faire le tour des composants principaux d'un système d'exploitation
- Donner une vue d'ensemble des différents environnements informatiques
- Explorer les systèmes d'exploitation libres

DIRO

Un système d'exploitation?

- Un programme qui fait l'intermédiare entre l'utilisateur d'un système informatique et le matériel

- Buts du système d'exploitation:
  - Exécuter les programmes de l'utilisateurs
  - Rendre le système informatique plus facile d'usage
  - Utiliser le matériel de manière efficace

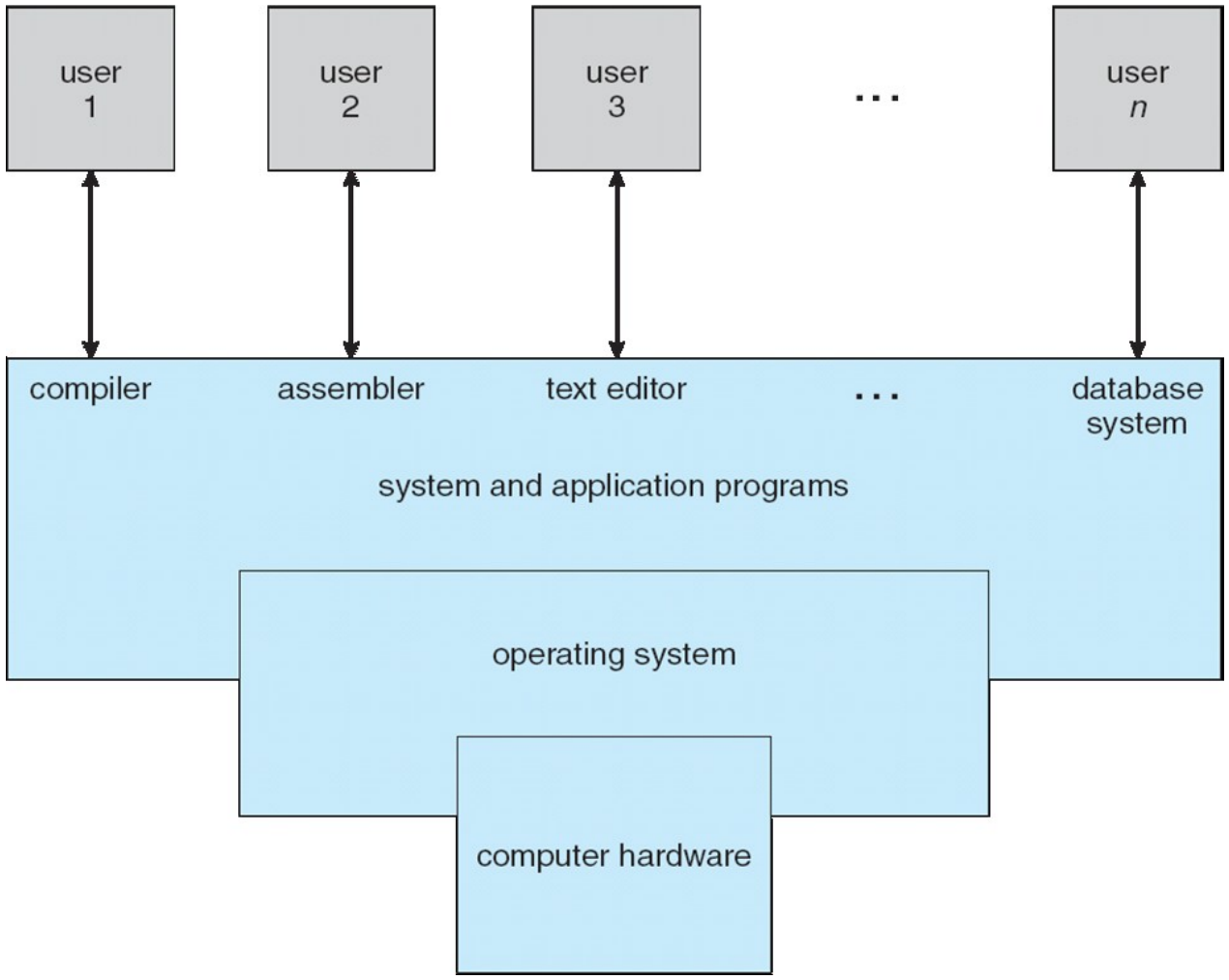
DIRO

Structure d'un système informatique

- Un système informatique peut se décomposer en 4 parties:
  - Matériel – fournis les resources informatiques fondamentales
    - CPU, mémoire, périphériques d'entrée/sortie (I/O)
  - Système d'exploitation
    - Contrôle et coordone l'usage du materiel entre les différents utilisateurs et applications
  - Programmes d'application – défini comment les resources du système sont utilisées pour résoudre les problèmes des utilisateurs
    - Traitements de textes, compilateurs, navigateurs, système de bases de données, jeux vidéo
  - Utilisateurs
    - Toi et moi, mais aussi d'autres machines

DIRO

Quatre éléments d'un système informatique



Que fait un système d'exploitation

- Ça dépend à qui on demande
- Sur un système partagé (serveur, **mainframe** ou **minicomputer**), partager les resources pour satisfaire tous les utilisateurs
- Utilisateurs de système dédiés (**workstations**, PC) partagent aussi souvent des resources sur un serveur
- Utilisateurs isolés veulent de la **facilité d'utilisation**
  - L'usage des resources est moins important
- Tablettes et téléphones ont des resources limitées, où il faut optimiser la durée de la batterie
- Certains ordinateurs n'ont pas d'interface uilisateur, tels que les systèmes embarqués dans des appareils ou des véhicules

DIRO

Définition d'un système d'exploitation

- SE est un **allocateur de resource**
  - Gère toutes les resources
  - Décide entre plusieurs requêtes en conflit, pour rendre l'usage des resources efficace et juste
- SE est un **programme de contrôle**
  - Contrôle l'exécution des programmes pour éviter les erreurs et usages incorrect de la machine

DIRO

Déf. d'un système d'exploitation (suite)

- Pas de définition acceptée universellement
- “Tout ce qu'un vendeur donne quand on lui demande un système d'e xploitation” est une bonne première approximation
  - Mais cela varie exagérément
- “Le ou les programmes qui sont toujours en fonctionnement”
- Le **noyau**. Tout le reste est soit un “programme système” (vient avec le système d'exploitation) ou un programme d'application

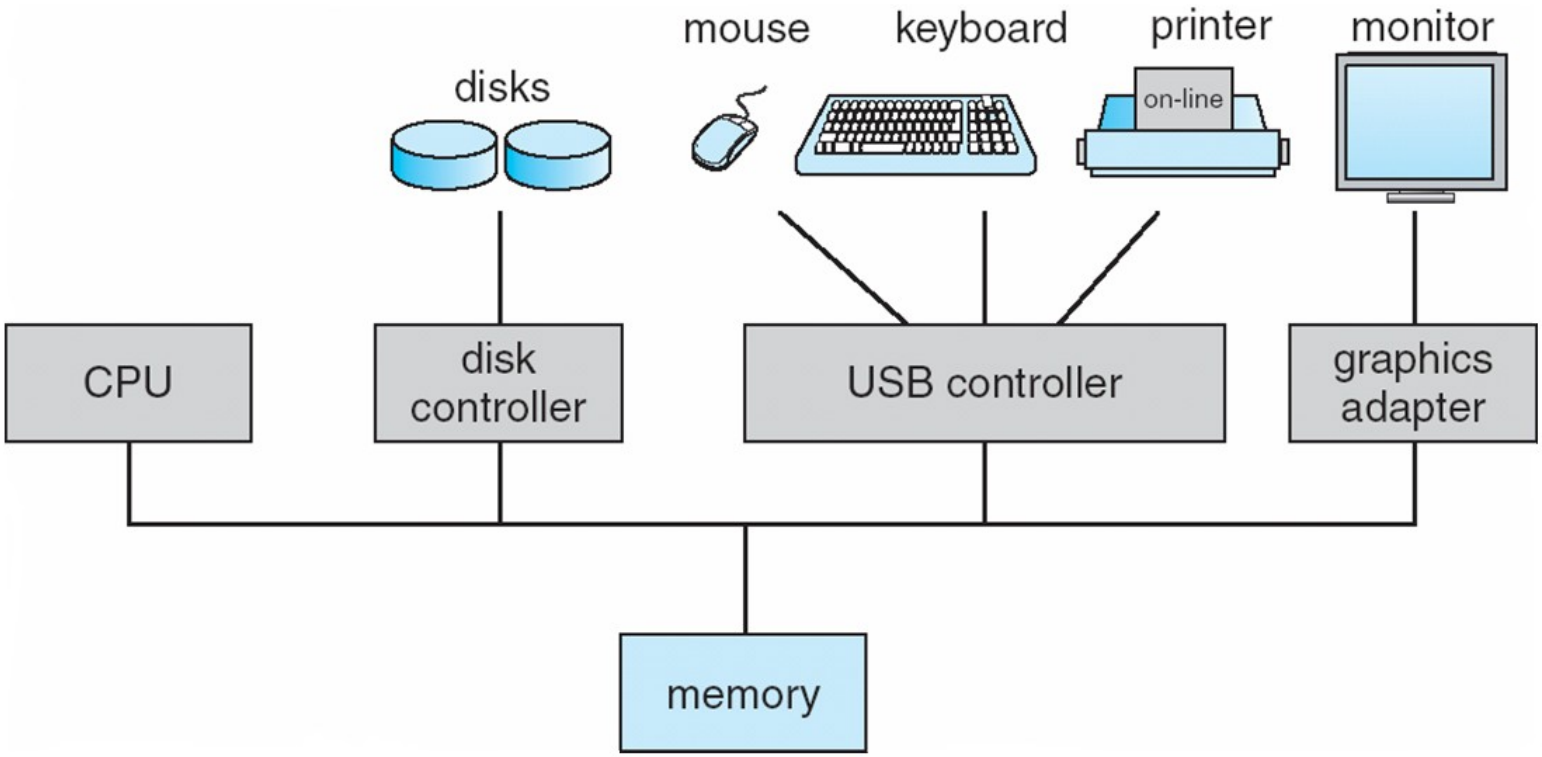
DIRO

Computer Startup

- **Programme de bootstrap** est chargé au démarrage
  - Habituellement stocké en mémoire Flash ou ROM (read-only memory), généralement dénommé **firmware**
  - Initialise tout le matériel du système
  - Charge le noyau du système d'exploitation et lance son exécution

Organisation d'un système informatique

- Opération d'un système informatique
  - Les CPUs et contrôleurs de périphériques sont connectés par un bus commun donnant accès à une mémoire partagée
  - Exécution concurrente des CPUs and des périphériques qui luttent pour accéder au bus et à la mémoire



Opération d'un système informatique

- CPUs et périphériques d'E/S peuvent exécuter concurremment
- Chaque contrôleur de périphérique se charge d'un type particulier de périphérique
- Chaque contrôleur de périphérique a un tampon local
- CPU déplace les données de/vers la mémoire centrale vers/de les tampons locaux
- Les E/S se font entre le périphérique et le tampon local du contrôleur
- Le contrôleur de périphérique informe le CPU qu'il a terminé une opération en signalant une interruption

Interruptions

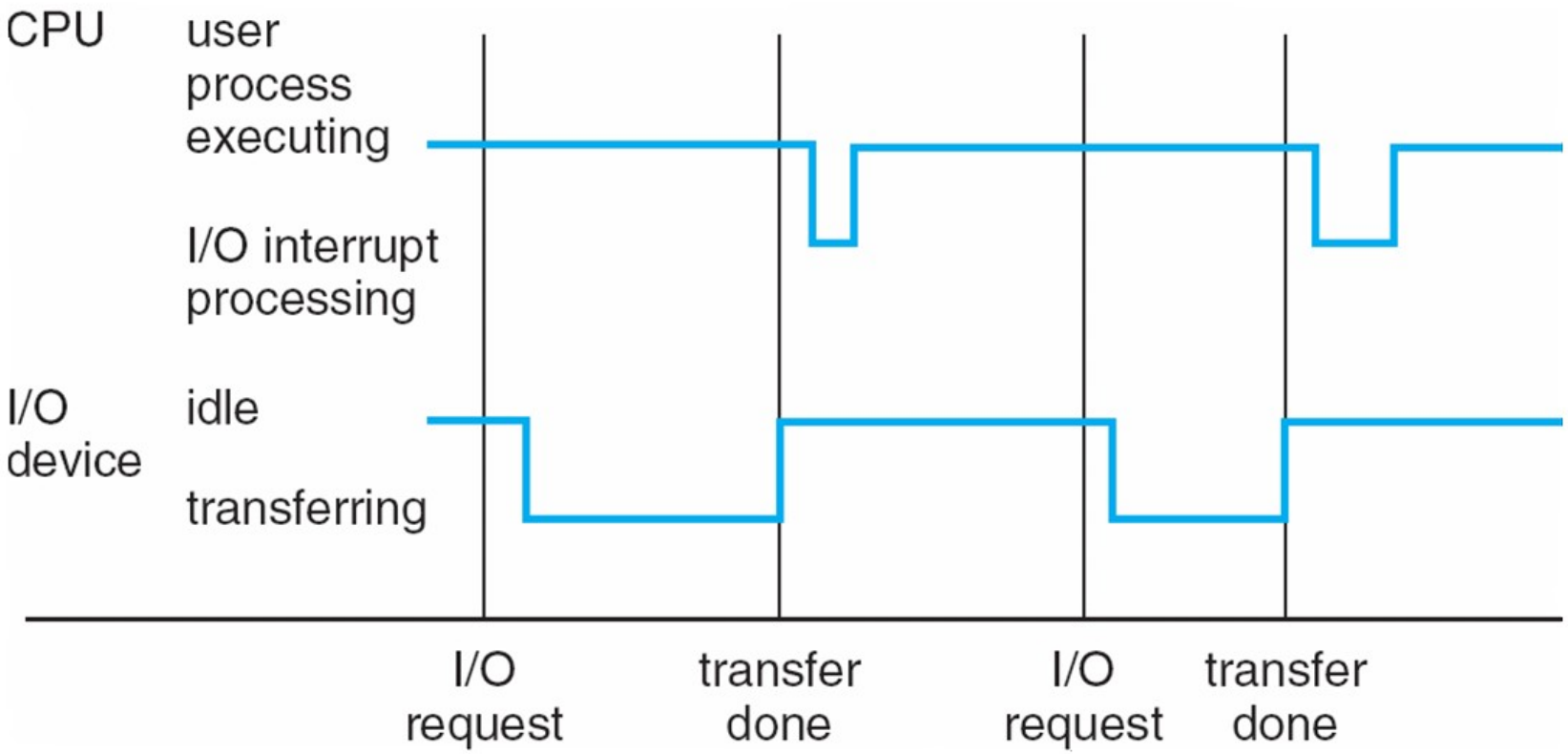
- Une interruption cause un transfert de contrôle à la routine de service d'interruptions, généralement via un **vecteur d'interruptions**, qui contient les adresses des différentes routines de service
- Le CPU sauve l'adresse de l'instructions interrompue
- Un “**trap**“ ou une **exception** est une sorte d'interruption générée par software causée par une erreur ou une requête
- Un système d'exploitation est **guidé par les interruptions**

DIRO

Gestion des interruptions

- Le système d'exploitation préserve l'état antérieur du CPU, tel que le contenu des registres
- Détermine quel type d'interrruption est arrivée
  - Par scrutation (**Polling**)
  - Par **vecteur d'interruption**
- Différent segments de code déterminent l'action à prendre pour chaque type d'interruption

Ligne de temps d'une interruption



DIRO

Structure des E/S

- Après le début d'une E/S, le contrôle retourne au programme seulement quand l'E/S est complétée:
  - Des instructions d'attente mettent le CPU en pause jusqu'à la prochaine interruption
  - Boucle d'attente active
  - Pas plus d'une requête d'E/S en attente à la fois, pas d'E/S simultanées

- Après le début d'une E/S, le contrôle retourne au programme sans attendre la fin de l'E/S:
  - **Appels systèmes** – requêtes au SE pour permettre au programme d'attendre la fin de l'E/S
  - **Table d'état des périphériques** contient le type, l'adresse, et l'état de chaque périphérique

DIRO

Storage Definitions and Notation Review

bit. Un bit contient soit 0 soit 1. Tout stockage est basé sur une collection de bits.

Un **byte** = 8 bits, et pour la majorité des systèmes, c'est la plus petite unité facile à manipuler. Par exemple, la majorité des CPU n'a pas d'instruction pour déplacer un bit, mais en a pour déplacer un byte.

b=bit; **B**=byte

Un **mot** est l'unité "naturelle" d'un système informatique. E.g. 32bit.

Un **kilobyte** (ou **KB**) = 1,024 bytes

Un **megabyte** (or **MB**) = 1,024^2 bytes

Un **gigabyte** (ou **GB**) = 1,024^3 bytes

Un **terabyte** (ou **TB)** = 1,024^4 bytes

Un **petabyte** (ou **PB**) = 1,024^5 bytes

Il est courant d'arrondir ces nombres en milliers, et de dire par exemple qu'un megabyte est 1 million de bytes and a gigabyte is 1 billion bytes. Les vitesses de communication sont souvent mesurées en multiple de bits plutôt que de bytes.

DIRO

Structure du DMA

DMA = Accès Direct à la Mémoire

- Utilisé pour les périphériques rapides, capables de transférer des données à haut débit
- Le contrôleur de DMA transfère des blocs de données directement de/vers la mémoire centrale sans intervention du CPU
- Une seule interruption par transfert plutôt qu'une pour chaque byte

DIRO

Structures de stockage

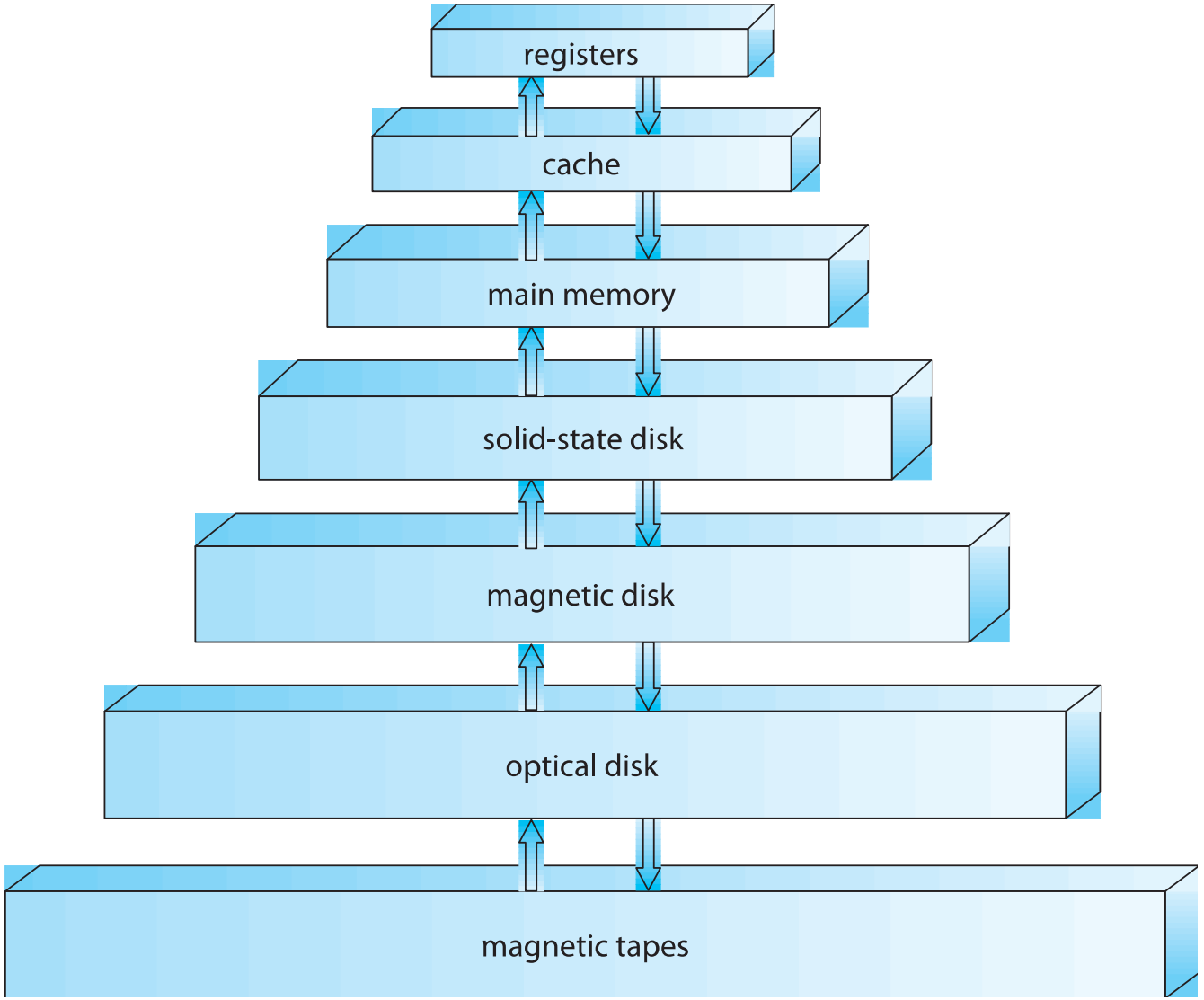
- Mémoire centrale – seul stockage auquel le CPU peut accéder directement
  - **Accès aléatoire**
  - Typiquement **volatile**
- Mémoire de masse – extension de la mémoire centrale qui offre une plus grande capacité et n'est **pas volatile**
- Disque magnétique – plateaux de metal ou verre recouvert d'un matériau magnétique d'enregistrement
  - La surface est divisée en **pistes**, elles mêmes subdivisées en **secteurs**
  - Le **contrôleur de disque** détermine l'interaction entre le périphérique et l'ordinateur
- **disques SSD** – plus rapides que les disques magnétiques mais quand même nonvolatils
  - Diverses technologies
  - Très populaire de nos jours

Hiérarchie de stockage

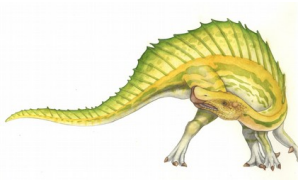
- Les systèmes de stockage sont organisés en hiérarchie
  - Vitesse
  - Coût
  - Volatilité
- **Caching** – copier l'information dans un système de stockage plus rapide; la mémoire centrale peut être utilisée comme un cache de la mémoire de masse
- **Pilote de périphérique** pour chaque contôleur de périphérique pour gérer les E/S
  - Fourni une interface uniforme entre le contrôleur et le noyau

DIRO

Hiérarchie de stockage



DIRO



Caching

- Principe important, utilisé à toute sortes d'endroits et de niveaux dans un ordinateur (en matériel, par le SE, par les applications)
- Copier l'information active vers une mémoire plus rapide
- Vérifier d'abord si la donnée est dans le cache
  - Si elle y est, on peut l'utiliser directement (rapide)
  - Si non, on la copie d'abord et on l'utilise après
- Pour être plus rapide, le cache est plus petit
  - Le design de la gestion du cache est très important
  - Taille du cache et règles de remplacement

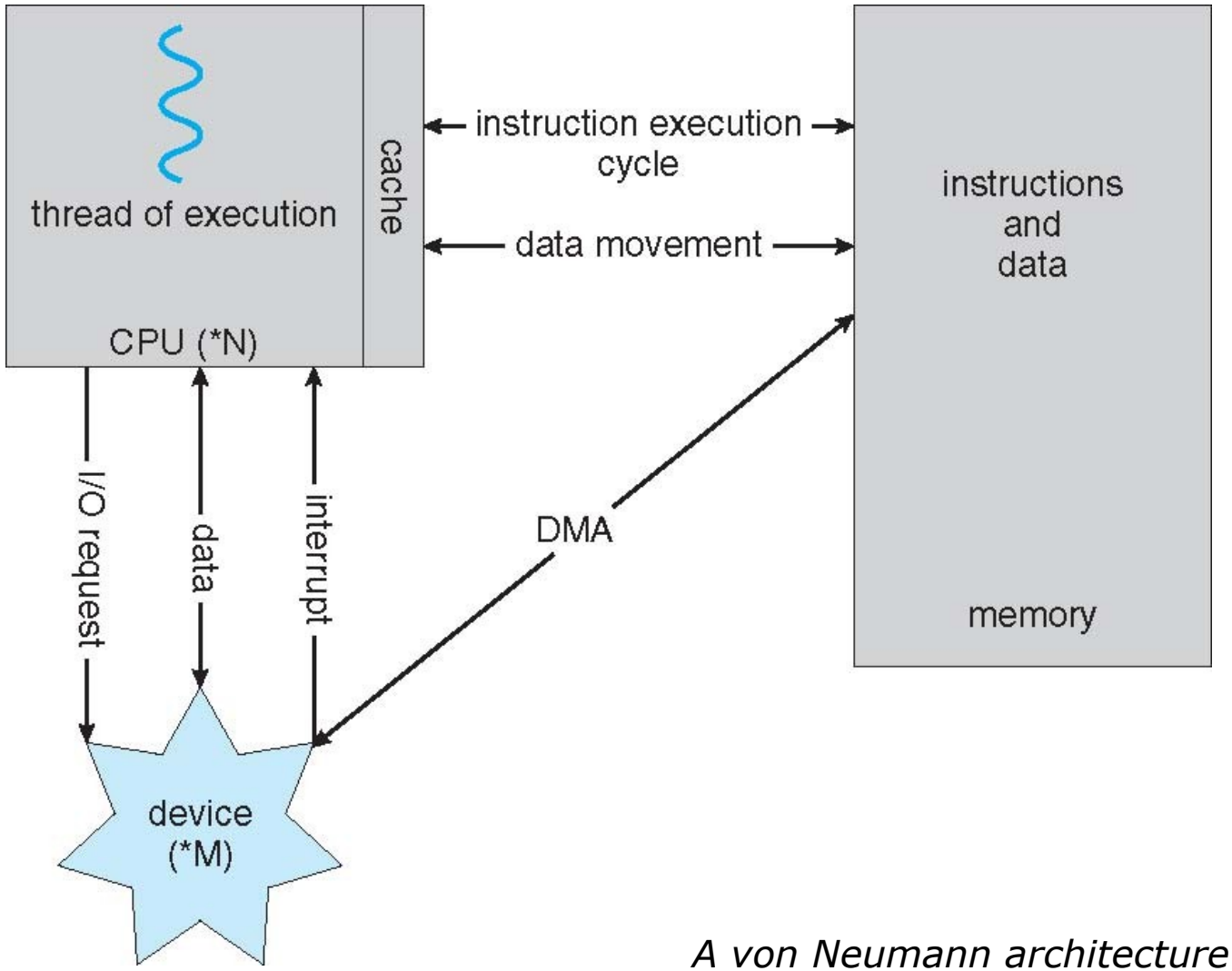
DIRO

Architecture des systèmes

- Beaucoup de système n'ont qu'un seul CPU
  - La majorité ont aussi des processeurs specialisés
- Les systèmes **multiprocesseurs** sont toujours plus courants et importants
  - Aussi nommés **systèmes parallels**, **systèmes fortement couplés**
  - Avantages:
    1. **Plus haute performance**
    2. **Économies d'échelle**
    3. **Meilleure fiabilité – dégradation progressive**, **tolérance aux pannes**
  - Deux types:
    - **Multiprocessing asymmétrique** (un CPU gère les autres)
    - **Multiprocessing symmétrique** (tous les CPUs sont égaux)

DIRO

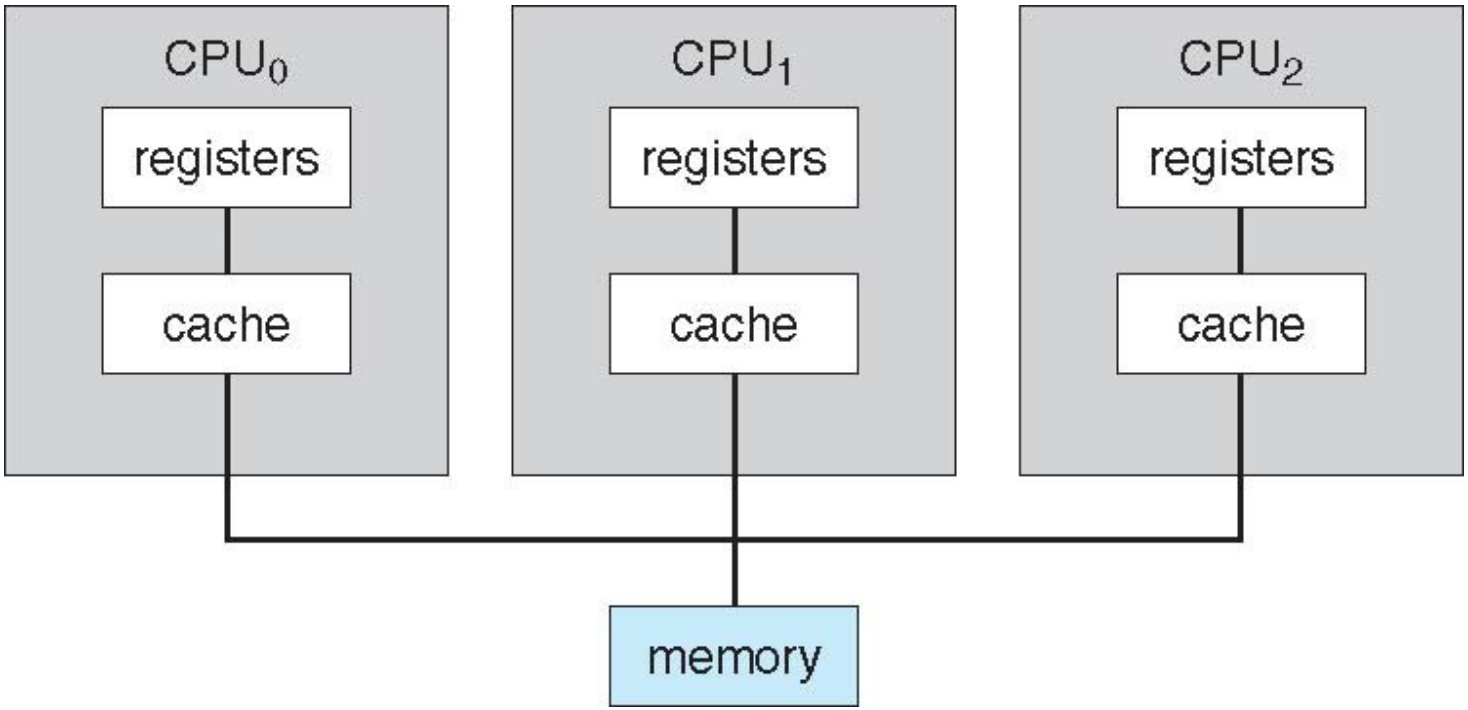
Comment fonctionne un ordinateur



A von Neumann architecture

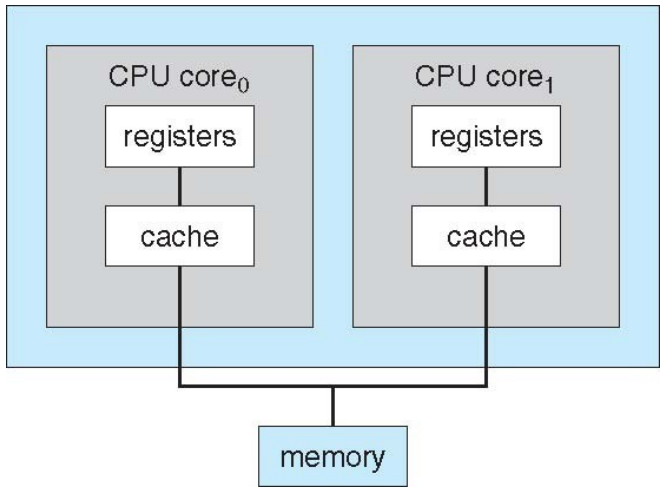
DIRO

Architecture multiprocesseur symmétrique



Un système dual-core

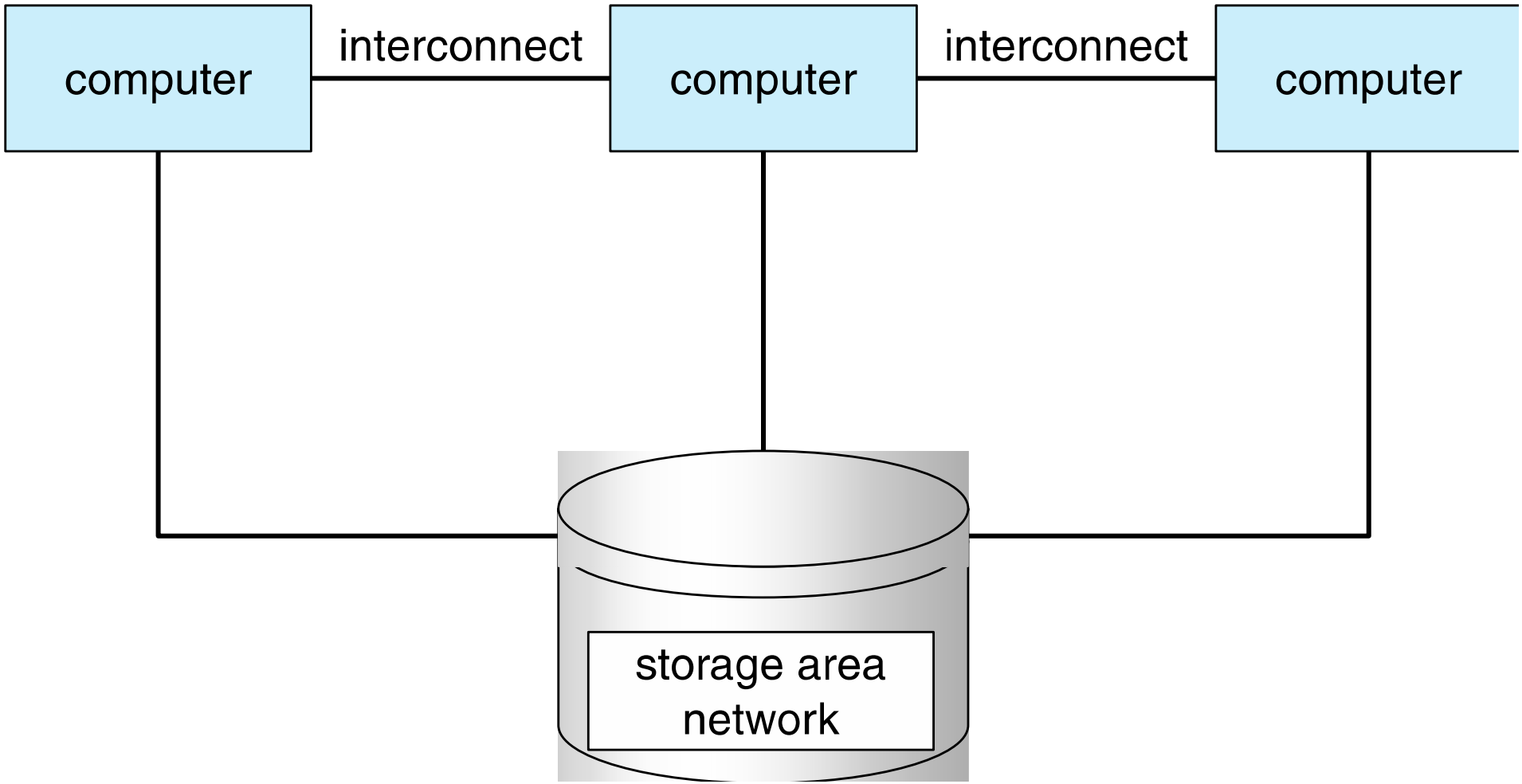
- Variations **UMA** et **NUMA** (accès mémoire uniforme et nonuniforme)
- Multi-chip ou **multicore**
- Systèmes contenant tout les chips vs **serveurs blade**
  - Chassis contenant plusieurs systèmes séparés (avec leur OS, I/O, réseau, etc.)



Systèmes en grappes

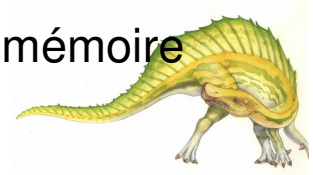
- Comme systèmes multiprocesseurs, mais multiples systèmes travaillant ensemble
  - Partage du stockage via un **storage-area network (SAN)**
  - Offre un service **high-availability** qui survit aux pannes
    - **Grappes asymmétriques** avec une machine en hot-standby
    - **Grappes symmétriques** où les nœuds se surveillent mutuellement
  - Grappes de calcul **high-performance computing (HPC)**
    - Applications écrites pour utiliser la **parallélisation**
  - Usage de **distributed lock manager** (**DLM**) pour éviter les conflits dans certaines opérations

Systèmes en grappes



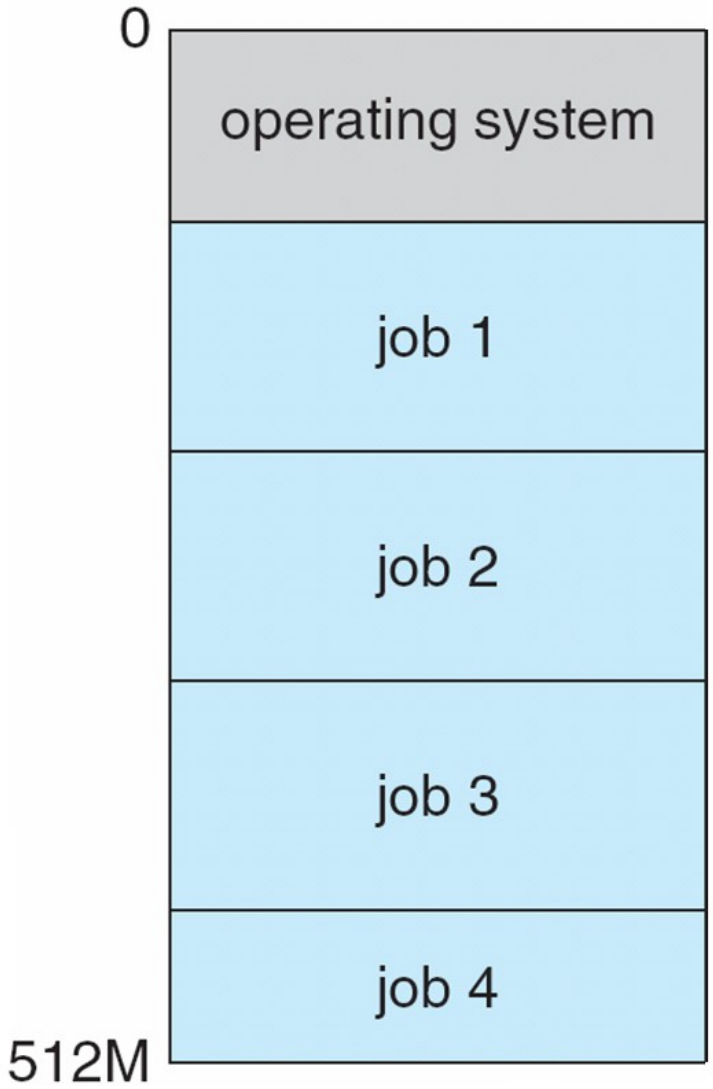
Structure des systèmes d'exploitation

- Besoin de **multiprogrammation** pour l'efficacité
  - Un seul utilisateur ne peut garder le CPU et les E/S occupées
  - Multiprogrammation organise le travail (code and data) pour que le CPU aie toujours quelque chose à faire
  - Un sous ensemble des tâches gardé en mémoire
  - Choix d'exécution des tâches par **job scheduling**
  - En cas d'attente (par exemple pour E/S), SE passe à une autre tâche
- **Timesharing** (**multitasking**) une extension naturelle où le CPU change de tâche si souvent que l'utilisateur peut interagir avec chacune d'elle: **interactive computing**
  - Temps de réponse devrait être < 1 second
  - Chaque utilisateur a au moins une tâche en mémoire **process**
  - Plusieurs tâches prêtes en même temps **CPU scheduling**
  - Tâches occupent plus de mémoire que disponible: **swapping** sur le disque
  - **Mémoire virtuelle** permet l'exécution de tâches partiellement en mémoire



DIRO

Layout mémoire de système multiprogrammé



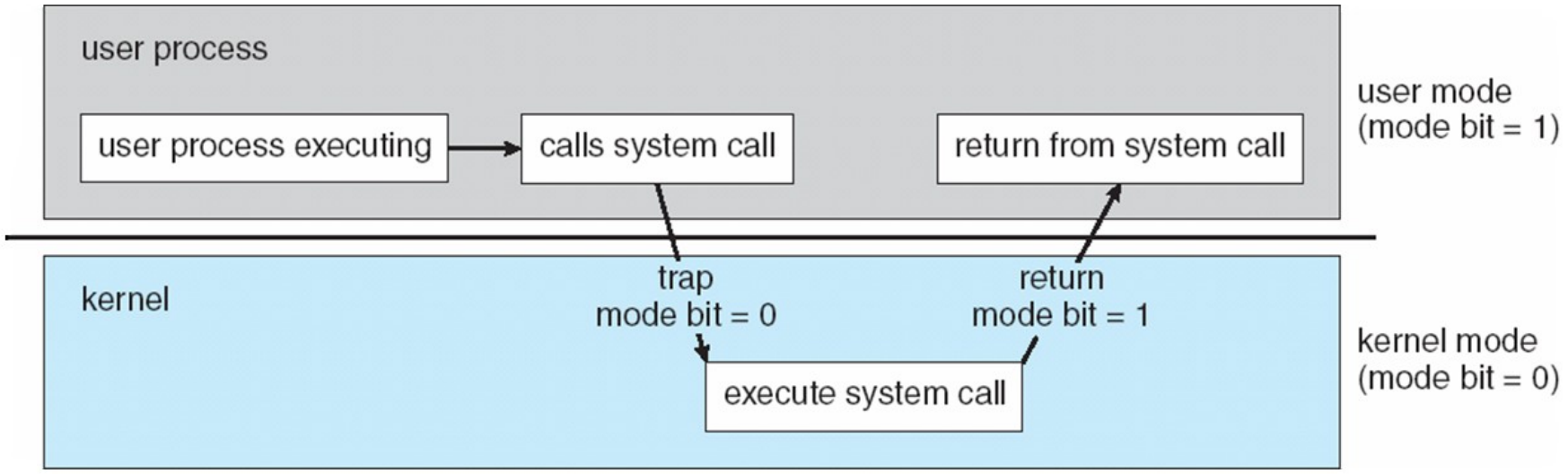
Opérations de systèmes d'exploitation

- **Guidé par les interruptions** par le matériel
- Erreur logicielle et requêtes créent des **exceptions ou trap**
  - Division par zéro, appel système
- Éviter les problèmes tels que les boucles infinies, les processus qui en modifient d'autres ou qui modifient le SE lui-même
- Opération en **deux modes** permet au SE de protéger les éléments
  - **Mode utilisateur** et **mode noyau**
  - **Bit de mode** fourni par le processeur
    - Permet de distinguer l'exécution dans le noyau
    - Les instructions “**privileged**”, autorisées seulement en mode noyau
    - Interruption et exceptions passent en mode noyau

DIRO

Transition au mode noyau

- Pour éviter les boucles sans fin, le noyau doit garder un œil ouvert
  - Utiliser un timer pour générer une interruption dans le futur
  - Avant de retourner en mode user, le SE arme le timer
  - Quand le timer sonne, l'interruption redonne la main au noyau



Gestion des processus

- Un processus est un programme en cours d'exécution. Un programme est une ***entité passive***, un processes est une ***entité active***.
- Unprocessus a besoin de resources pour accomplir sa tâche
  - CPU, mémoire, E/S, fichiers
  - Initialisation des données
- À la fin d'un processus, il faut récupérer ces resources
- Un processus “single-threaded” a un **program counter** qui indique l'adresse de la prochaine instruction à exécuter
  - Le processus exécute ses instructions séquentiellement, une à la fois, jusqu'à épuisement
- Un processus “multi-threaded” a plusieurs program counters, un par thread, qui peuvent ainsi exécuter en parallèle
- Habituellement un système a toujours beaucoup de processus actifs, certains utilisateurs, d'autres appartenant au système, qui fonctionnent tous concurremment sur un ou plusieurs CPUs
  - Multiplexer les CPUs parmis les processus / threads

DIRO

Activités de gestion des processus

Le système d'exploitation est responsable de:

- Créer et détruire les processus utilisateurs et système
- Suspendre et réveiller les processus
- Fournir des méchanismes pour la synchronisation entre processus
- Fournir des méchanismes pour la communication entre processus
- Fournir des méchanismes pour la gestion des étreintes mortelles

DIRO

Gestion mémoire

- Avoir les données en mémoire avant de les manipuler
- Avoir les instructions en mémoire avant de les exécuter
- La gestion mémoire détermine quoi garder en mémoire pour
  - Optimiser l'usage du CPU et le temps de réponse à l'utilisateur
- Activités de gestion mémoire
  - Garder trace de qui utilise quelle partie de la mémoire
  - Decider quelles informations de quels processus transférer de/vers la mémoire
  - Allouer et désallouer les espaces mémoire au besoin

DIRO