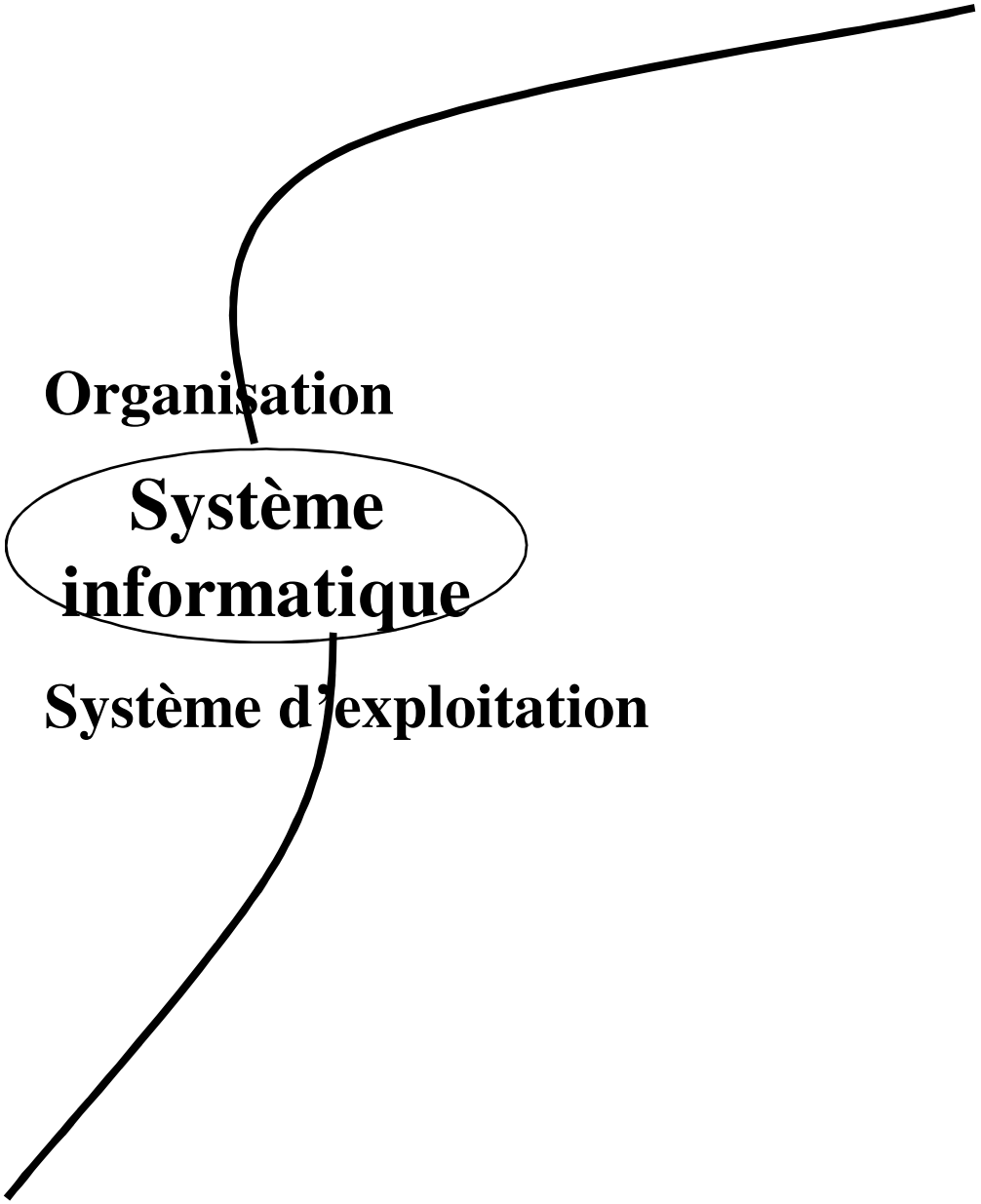
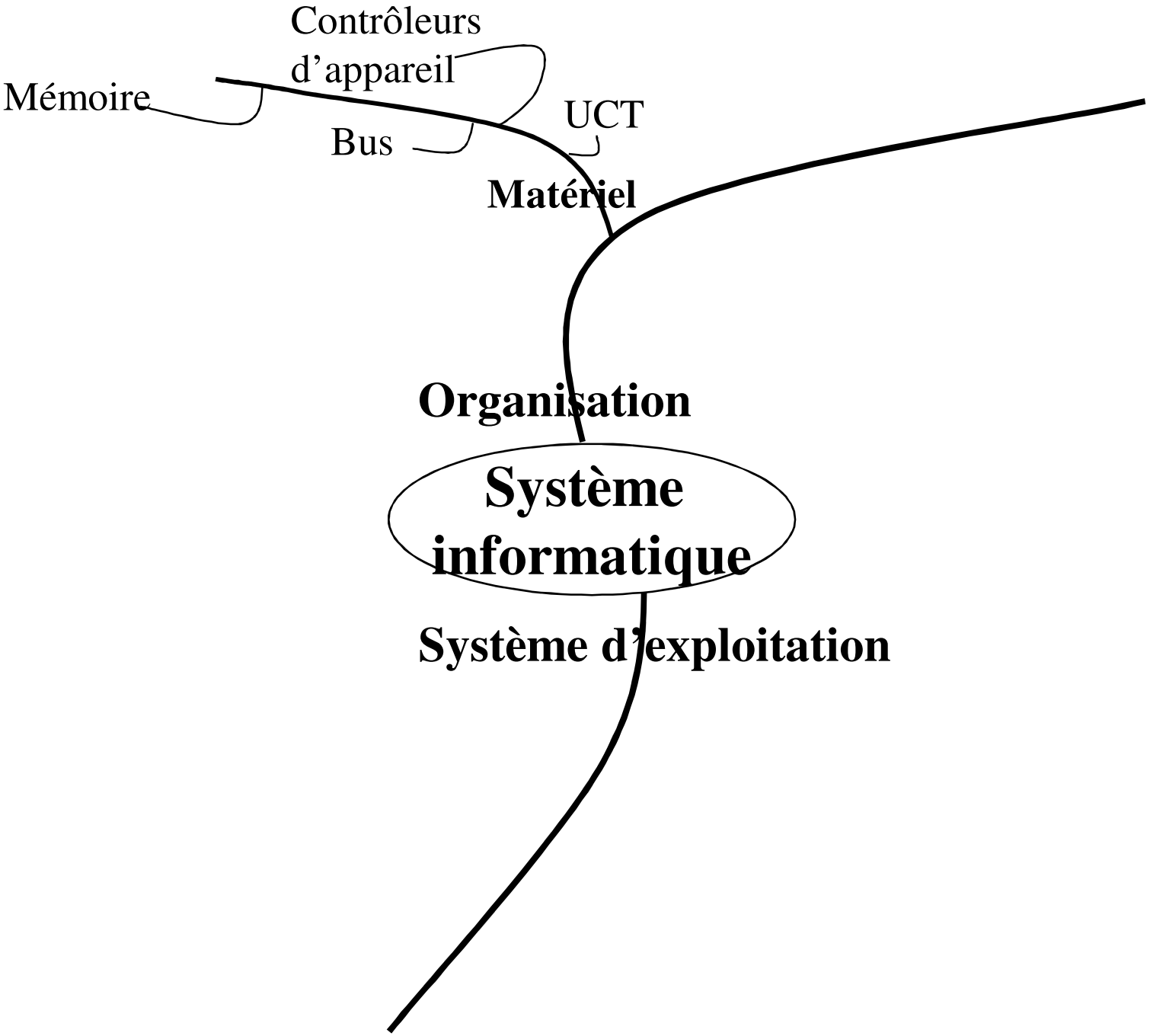
CSI3531 Chapitre 1 - Introduction/survol du SE

Lecture: Chapitre 1 et 2 (Silberchatz)

Objectif:

- Faire un survol rapide de l'organisation des ordinateurs – le processeur (UCT), la mémoire, et le système d'entrée/sortie, l'architecture et les opérations générales.
- Introduire le système d'exploitation afin de comprendre son rôle et ses fonctions principales

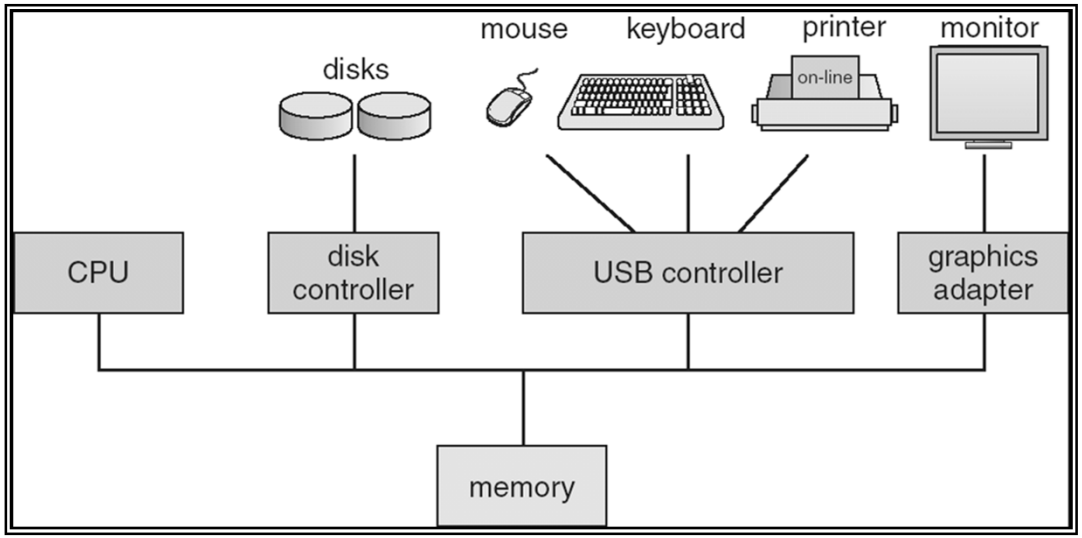

Organisation
Système informatique
Système d'exploitation

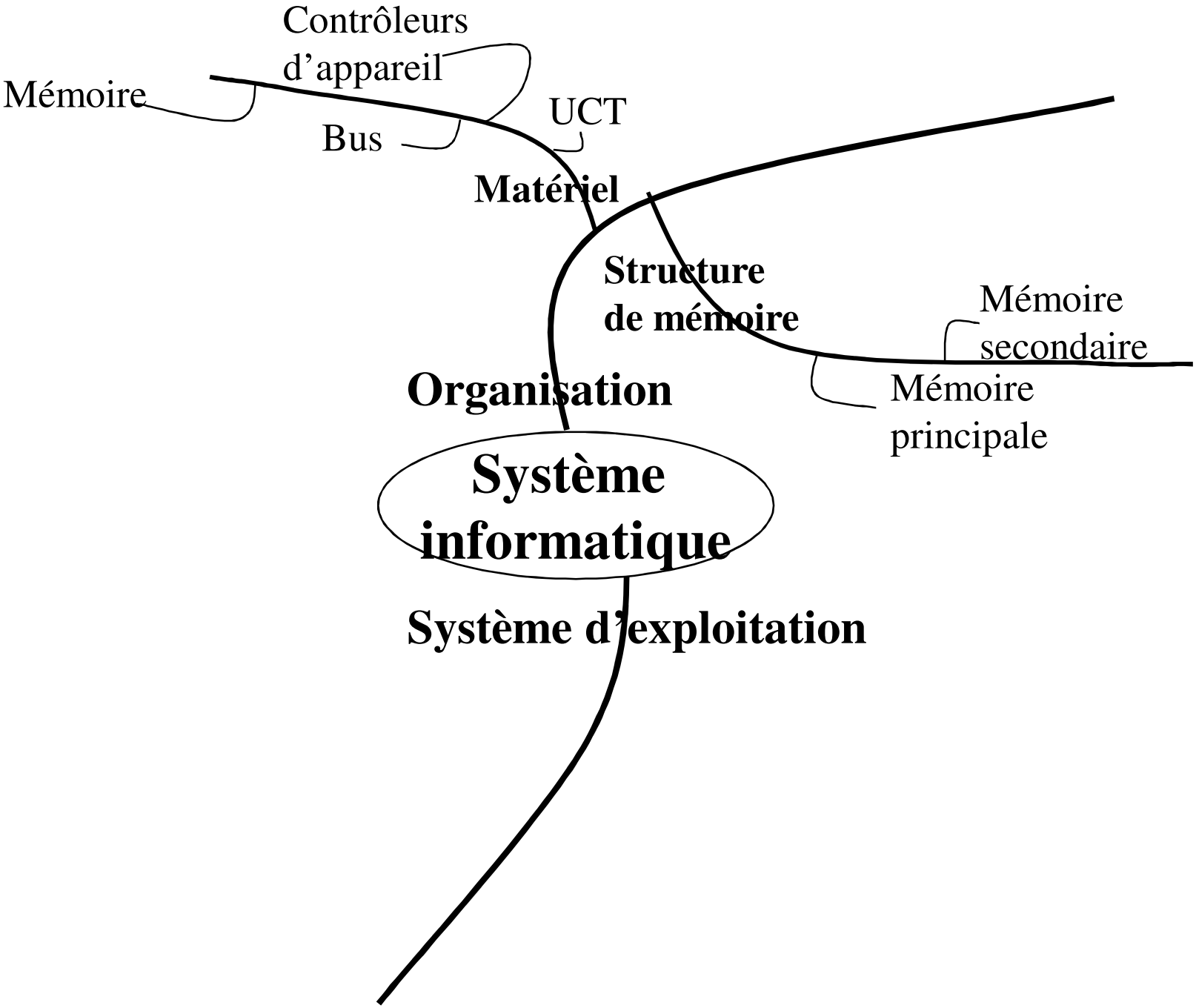

Contrôleurs
d'appareil
Mémoire
UCT
Bus
Matériel
Organisation
Système
informatique
Système d'exploitation

Matériel principal

- Le Processeur (UCT)
- La Mémoire principale (mémoire réelle, RAM)
  - Contient le code et les données
- Les Modules E/S (Contrôleurs E/S, processeurs E/S...)
  - Le matériel (avec registres: ports E/S) de transport des données entre UCT et périphériques comme:
    - La mémoire secondaire (ex: disques rigides)
    - Le clavier, écran...
    - L'Équipement de communication

- L'Interconnexion (ie: Bus)
  - permet la communication entre le(s) processeur(s), la mémoire et les modules E/S



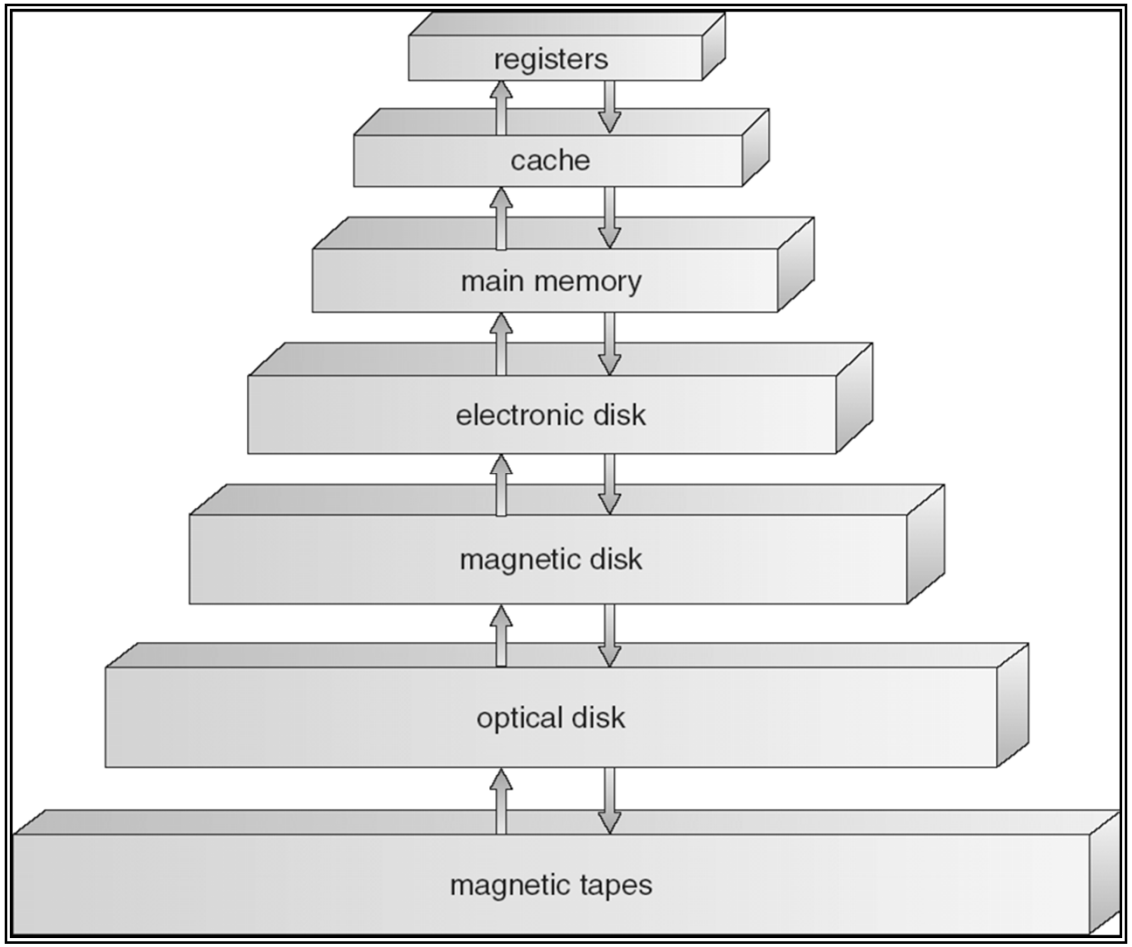

Contrôleurs
d'appareil
Mémoire
UCT
Bus
Matériel
Structure
de mémoire
Mémoire
secondaire
Mémoire
principale
Organisation
Système
informatique
Système d'exploitation

Structure de stockage

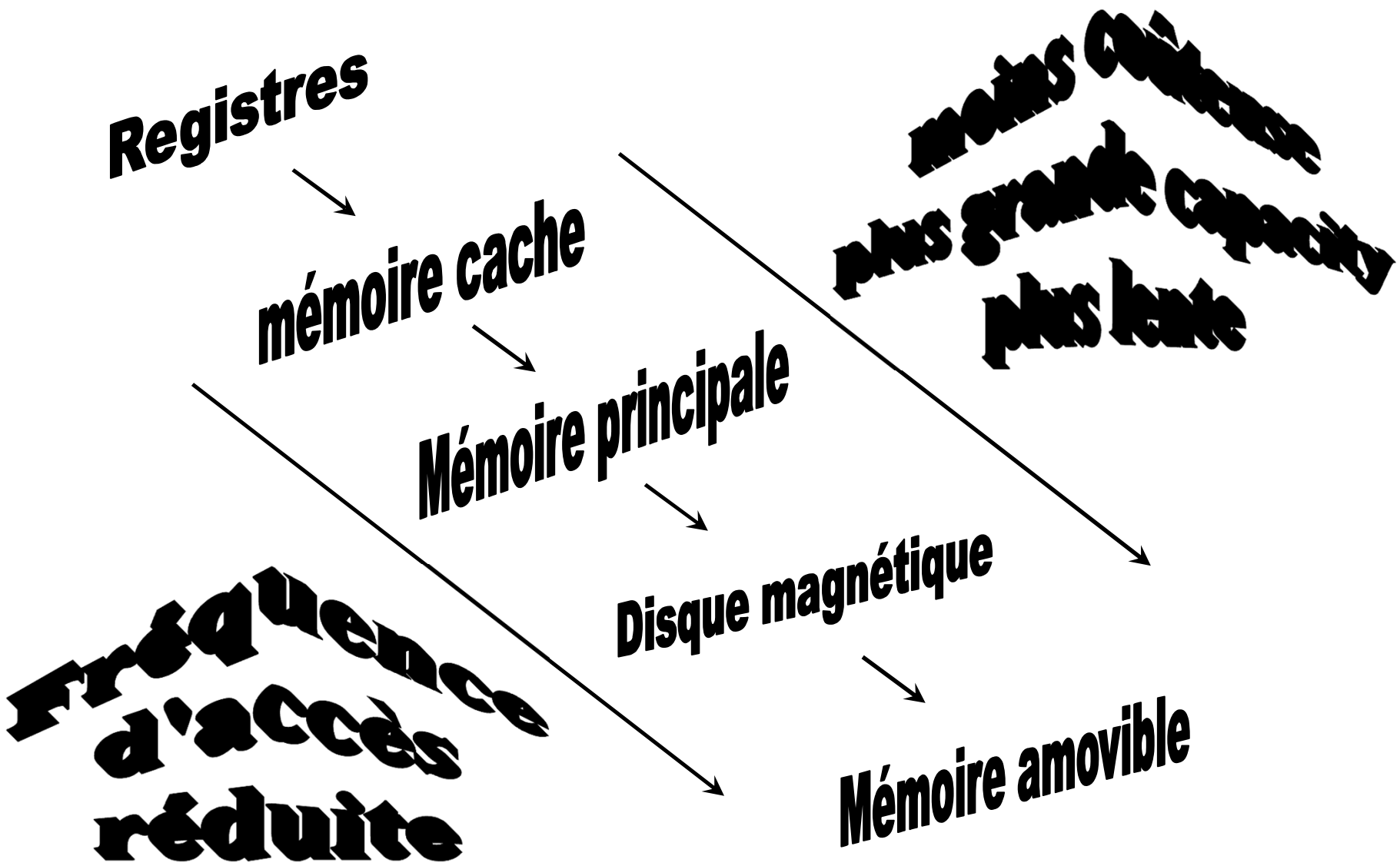
- Le stockage primaire
  - **Accessible directement par l'UCT**
  - **Un programme doit être dans la mémoire principale pour être exécuté par l'UCT**
  - **La mémoire principale n'est pas assez grande pour contenir tous les programmes et données**
  - **La mémoire principale est *volatile* – son contenu change avec toute perte de puissance et au redémarrage.**
- Le stockage secondaire
  - Il contient, de façon permanente, de grandes quantités de données/fichiers.

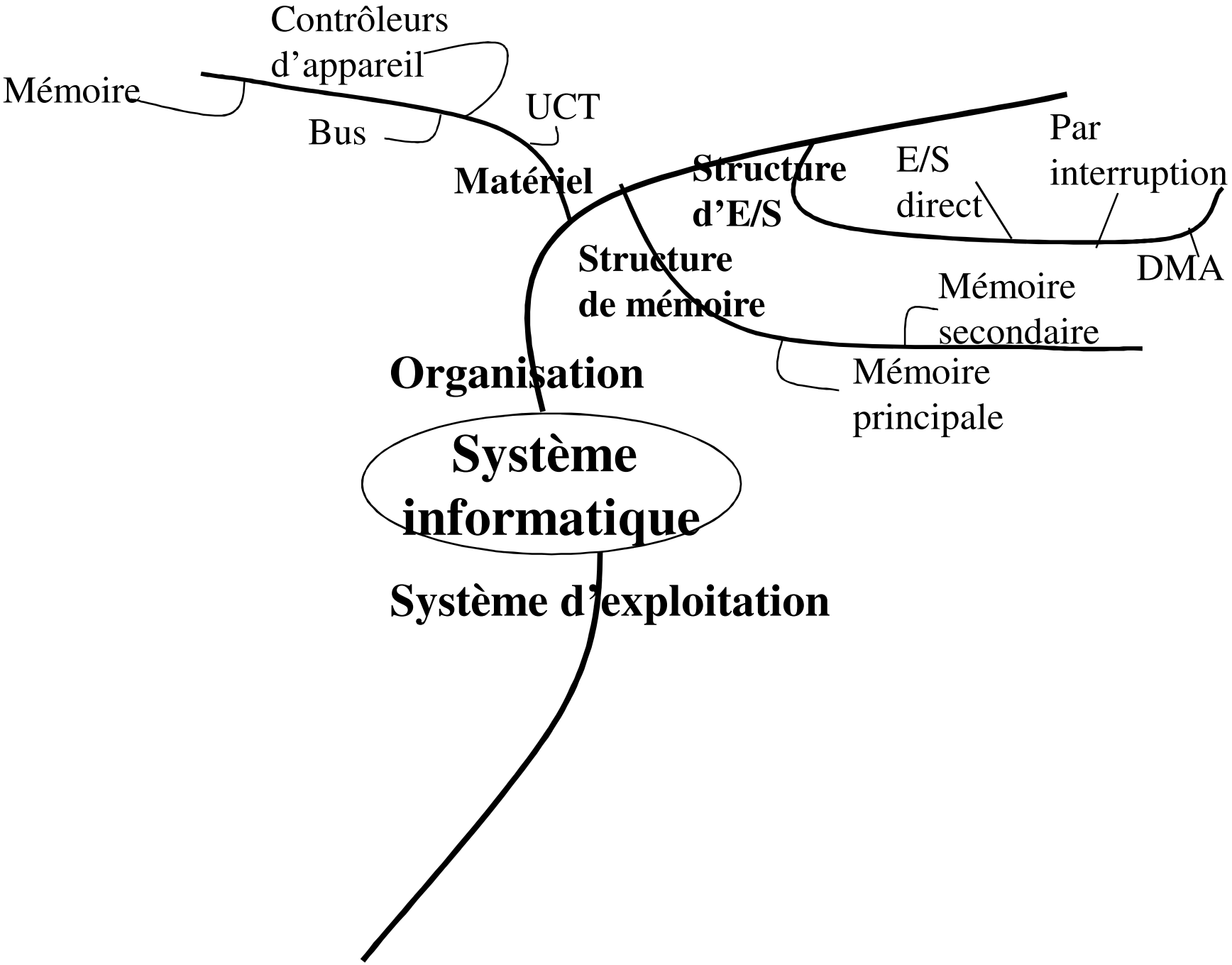
- **En générale, il existe une hiérarchie qui régit les différents types de mémoire, elle varie selon la vitesse, le coût, la taille et la volatilité.**

Hiérarchie du système de mémoire

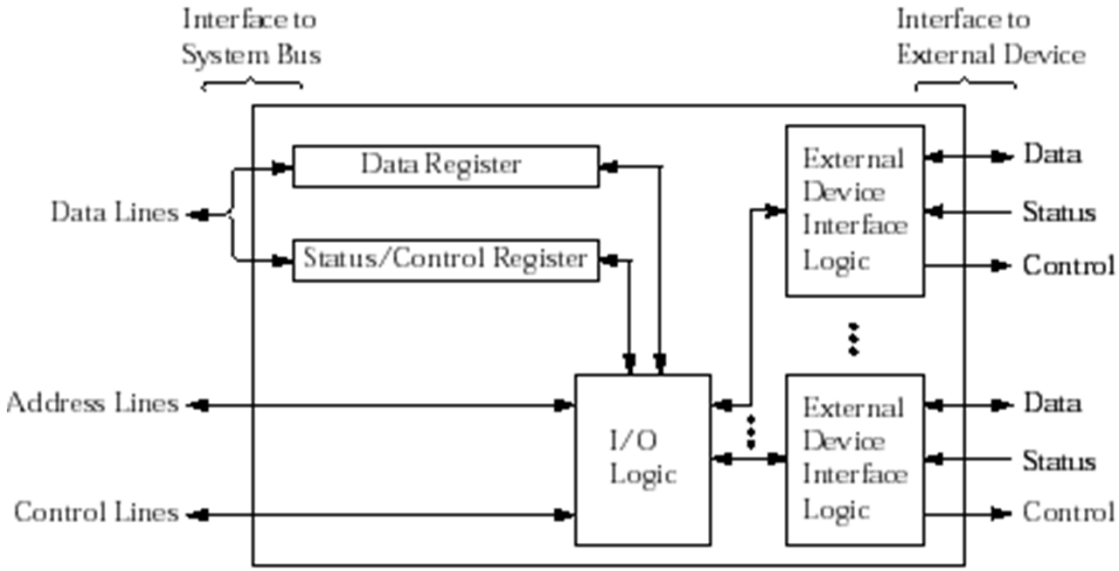


Organisation hiérarchique de la mémoire




Contrôleurs
d'appareil
Mémoire
UCT
Bus
Matériel
Structure
d'E/S
E/S
direct
Par
interruption
DMA
Structure
de mémoire
Mémoire
secondaire
Mémoire
principale
Organisation
Système
informatique
Système d'exploitation

Structure du contrôleur E/S

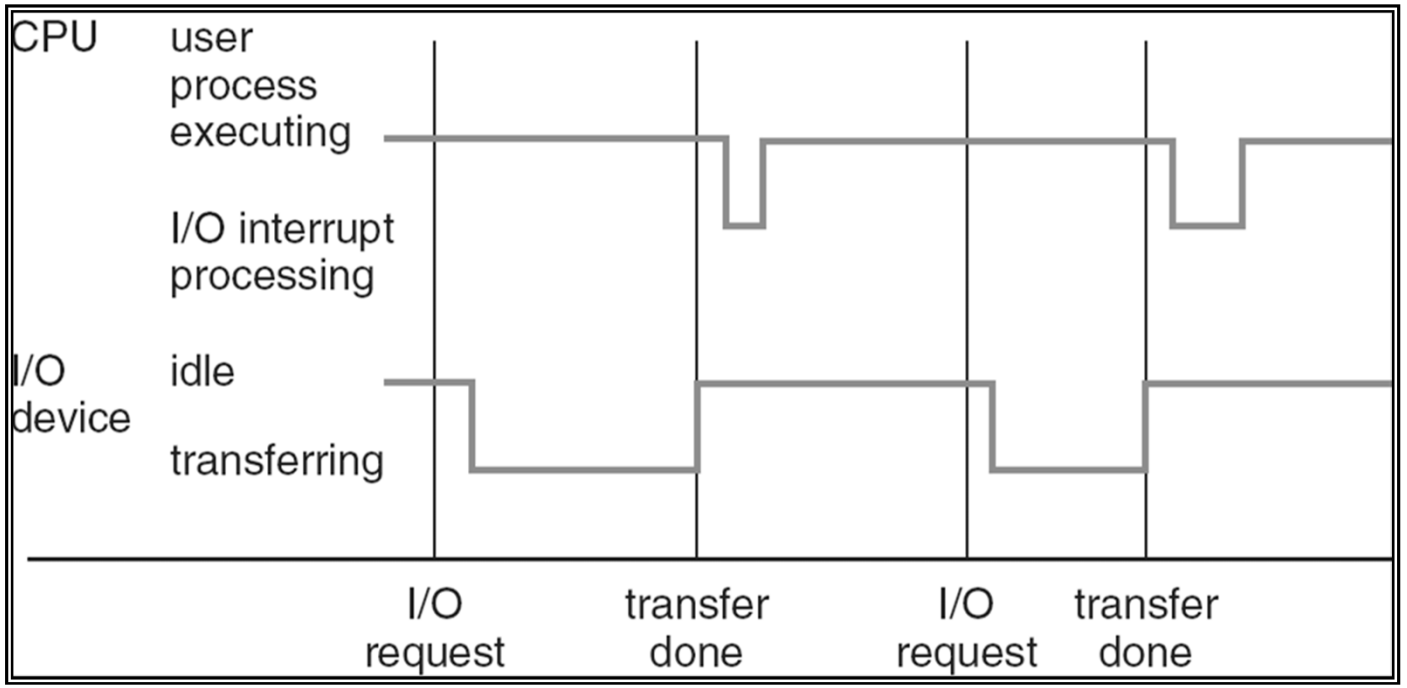


- **Les Données du bus sont stockés dans le(s) registre(s) de données "data register" (ports E/S)**
- **Le registre "Status/Control" contient:**
  - **l'information du statut de l'opération E/S**
  - **l'information de contrôle provenant de l'UCT**
- **Le circuit "I/O logic" interagit avec l'UCT via le bus.**
- **Il contient la logique spécifique de l'interface de chaque dispositif**

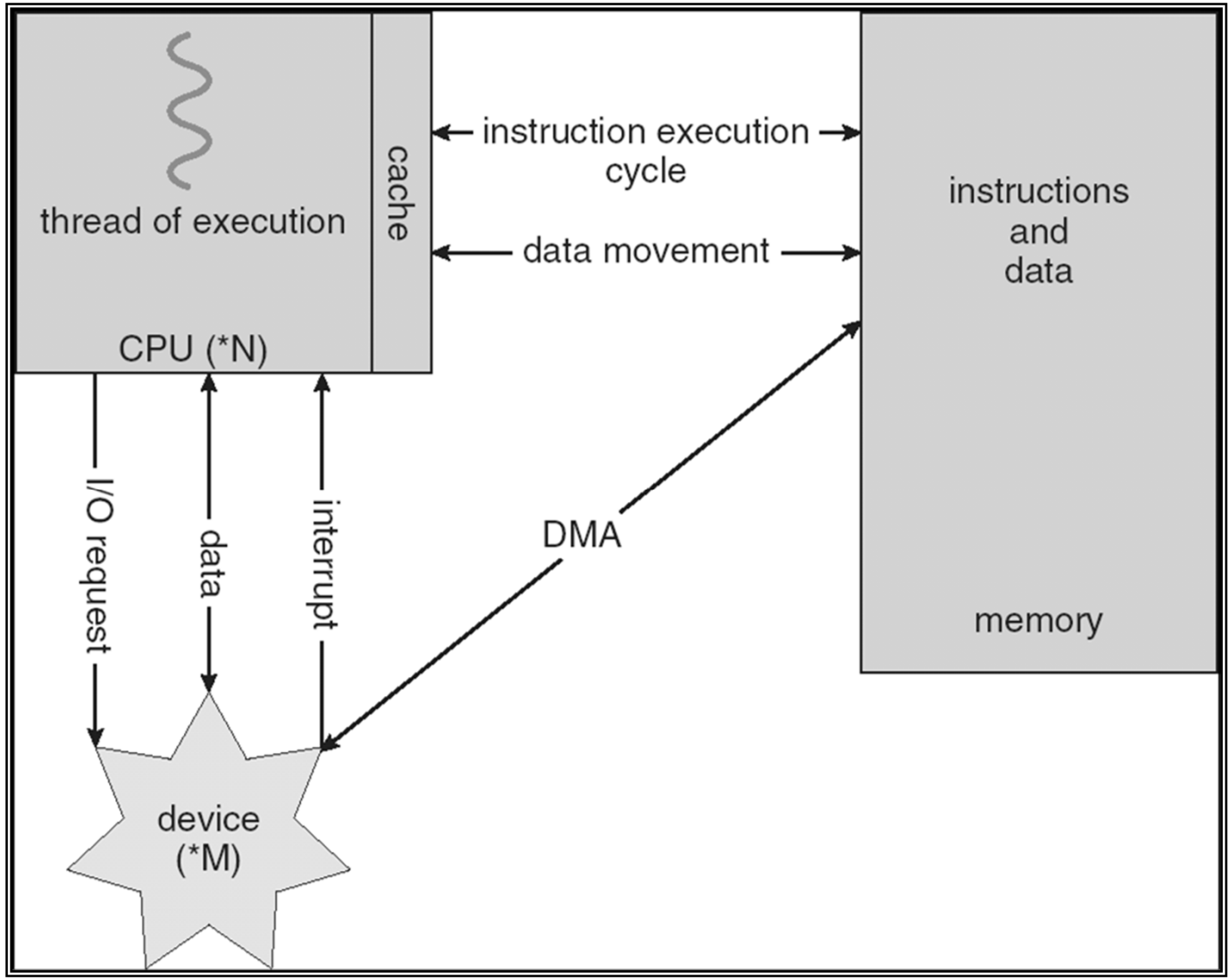
Techniques de communication des E/S

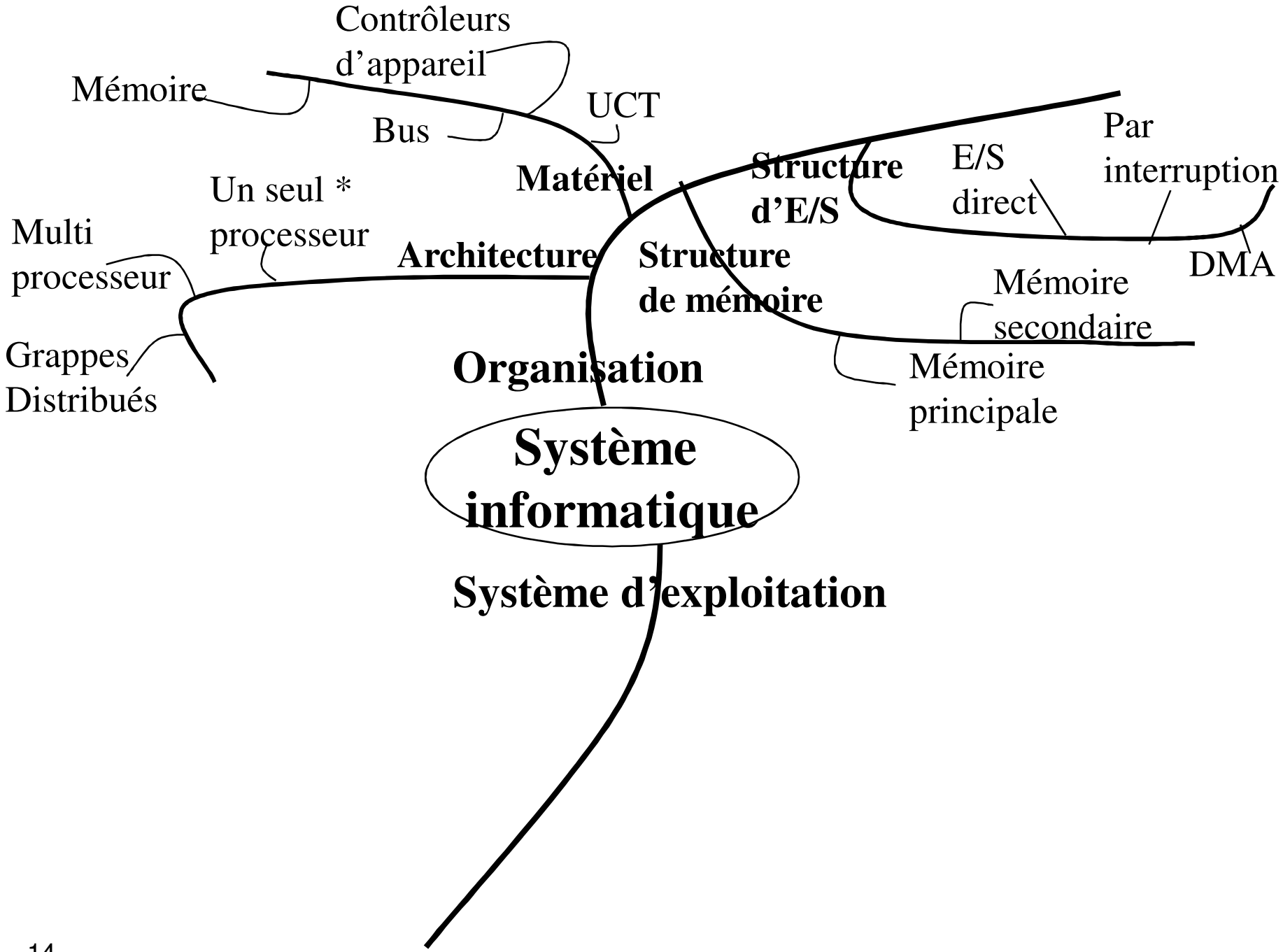
- **3 techniques sont disponibles:**
  - **La méthode d'E/S programmées**
    - **N'utilise pas d'interruptions. L'UCT doit attendre après chaque opération d'E/S**
  - **La méthode d'E/S déclenchées par interruptions**
    - **L'UCT peut exécuter des instructions pendant l'opération E/S: il est interrompu lorsque cette dernière est terminée.**
  - **Accès direct à la mémoire (DMA)**
    - **Un bloc de données est transféré directement dans ou de la mémoire sans passer par l'UCT**

Interruption de l'E/S



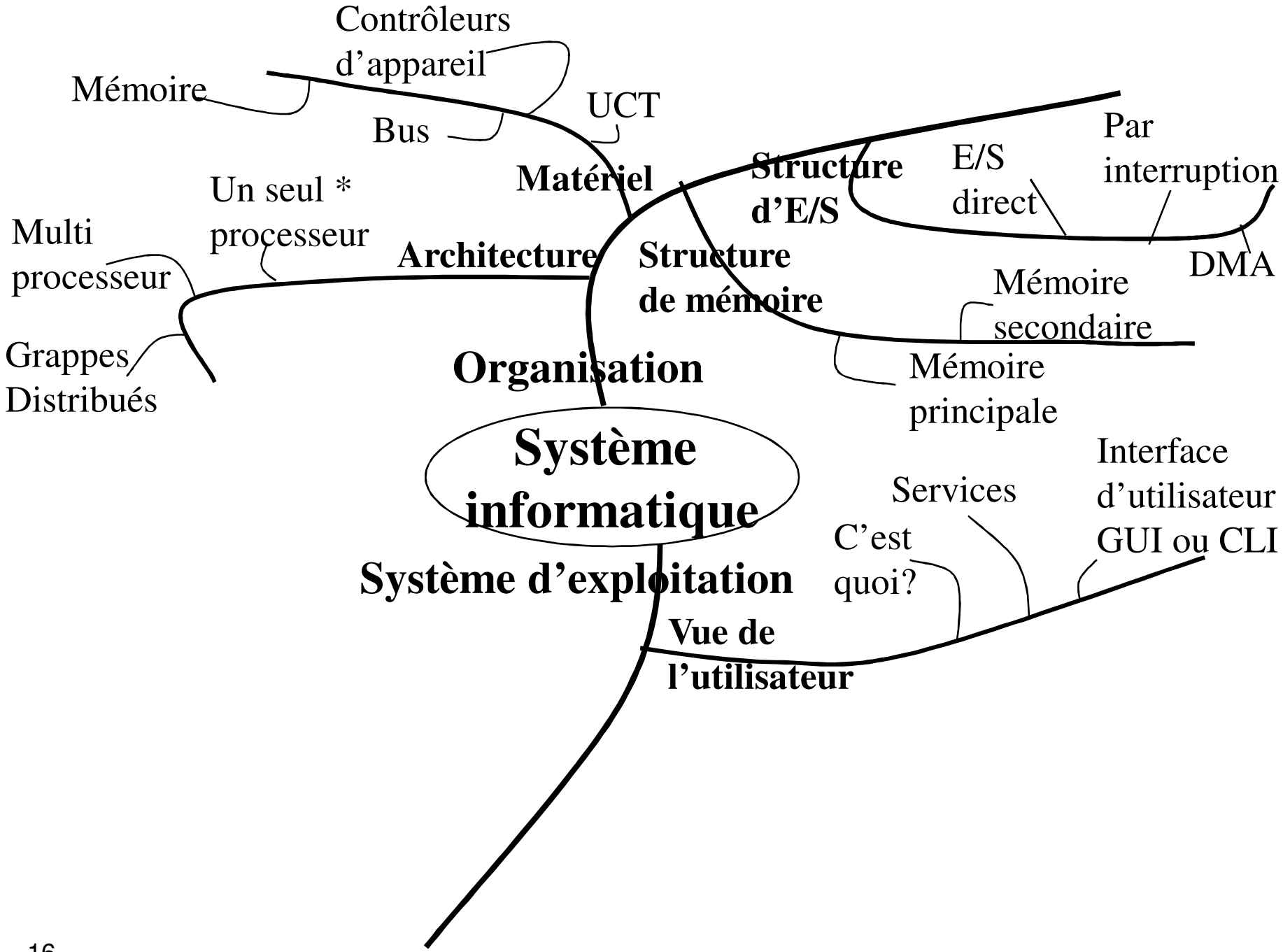
Fonctionnement de l'ordinateur moderne



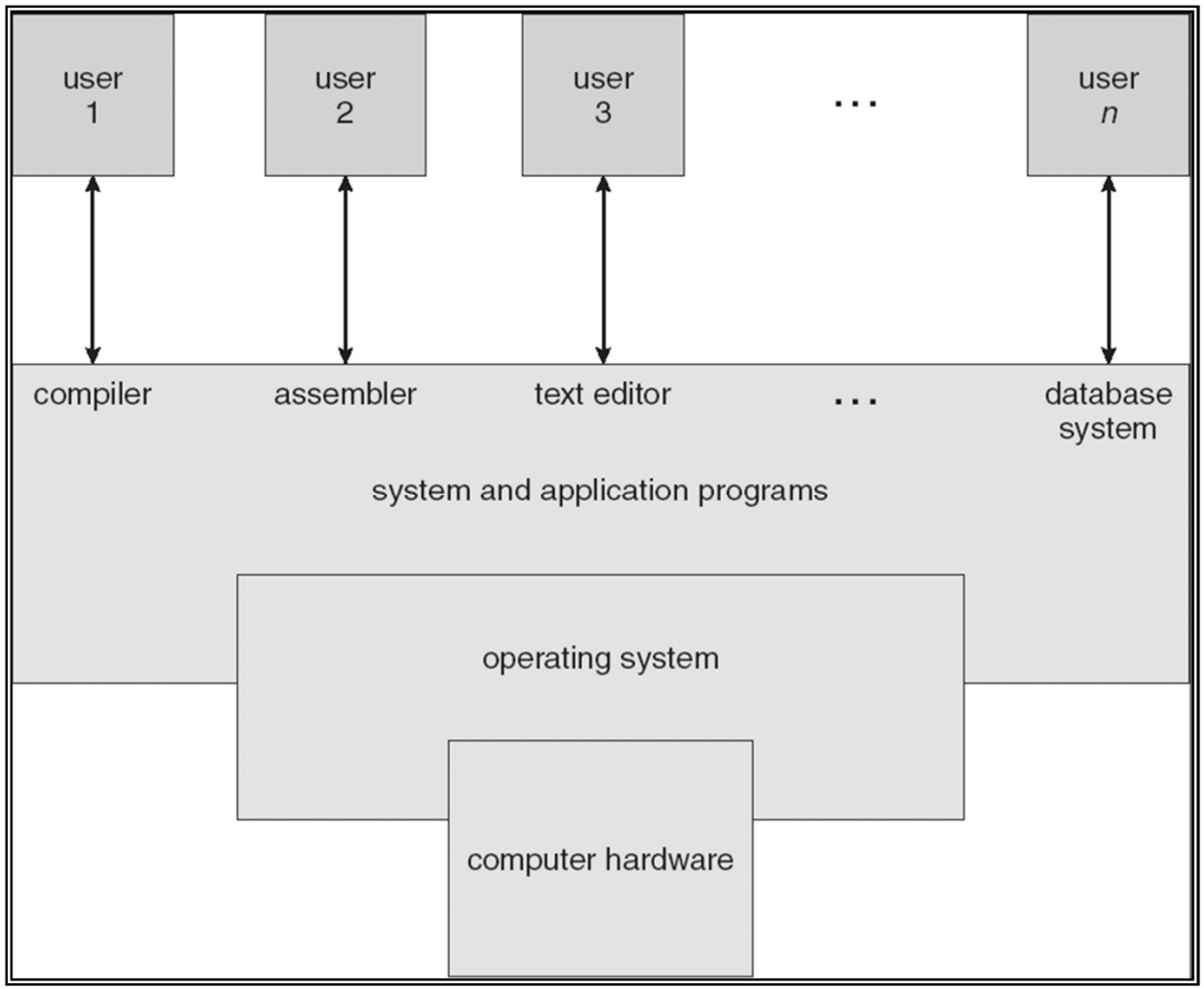

Contrôleurs d'appareil
Mémoire
Bus
UCT
Matériel
Un seul * processeur
Multi processeur
Grappes
Distribués
Architecture
Organisation
Structure de mémoire
Structure d'E/S
E/S direct
Par interruption
DMA
Mémoire secondaire
Mémoire principale
Système informatique
Système d'exploitation

Architecture des systèmes informatiques

- Systèmes à un seul processeur
  - Du PDA à l'ordinateur central
  - Presque tous on des processeurs spécialisé pour graphisme et E/S
    - Ceci n'est pas considéré comme un multiprocesseur
- Systèmes multiprocesseurs
  - Débit de traitement accru
  - Économies d'échelle
  - Fiabilité accru
  - Multitraitement asymétrique
    - Chaque processor est attribué une tâche spécifique
  - Multitraitement symétrique (le plus commun)
    - Tous les processeurs accomplissent toutes les tâches du SE
- Grappes, systèmes distribués


Contrôleurs d'appareil
Mémoire
Bus
UCT
Matériel
Un seul * processeur
Multi processeur
Grappes
Distribués
Architecture
Structure d'E/S
E/S direct
Par interruption
DMA
Structure de mémoire
Mémoire secondaire
Mémoire principale
Organisation
Système informatique
Système d'exploitation
Vue de l'utilisateur
C'est quoi?
Services
Interface d'utilisateur GUI ou CLI

Vue abstraite des composantes d'un système informatique



Perspectives de l'utilisateur d'un ordinateur

- Ceci est mon système, il n'y a que moi qui l'utilise
  - **i.e. PC dont un utilisateur monopolise**
  - **Le SE maximise le travail (ou le jeu) de l'utilisateur**
  - **Le SE est conçu à l'origine pour facilité son utilisation et non pas pour l'utilisation de ressources.**
  - **Les systèmes portatifs – petite demande au niveau matériel**
- Du point de vue d'accès aux ressources, je suis chanceux d'avoir du temps d'UCT
  - i.e l'ordinateur central ou le mini-ordinateur
  - L'ES est conçu pour maximiser l'utilisation des ressources (UCT, mémoire, E/S)
- Le partage des ordinateurs
  - i.e. les stations de travails branchées à un réseau de servers
  - Ressources dédiées et partagées
  - ES balance les besoins individuelles avec les besoins d'utilisation des ressources
- Quoi? Il y a un ordinateur à l'intérieur.
  - Systèmes imbriqués conçu pour rouler avec un minimum d'intervention

Pourquoi des systèmes d'exploitation?

Quand il y a plusieurs utilisateurs/programmes desservis:

- **Le partage des ressources**
- **Qui obtient les ressources? quand?**
- **Quel utililisateur ou application est permis de faire quoi?**
- **Comment facturer les utilisateurs?**

  Ok, ok, on doit gérer plusieurs utilisateurs/programmes, mais qu'arrive-t-il avec un seul utilisateur? (i.e. pour mon PC)
- **Abstraction du matériel**
- **On veut toujours pouvoir exécuter plusieurs programmes concurremment.**

Que font les systèmes d'exploitations?

- **Donnez une ou deux phrase qui résume le role du SE.**
  - **Le SE est le programme le plus impliqué avec le matériel**
    - **Abstraction du matériel**
  - **Le SE s'occupe de l'allocation des ressources**
    - **Gère toutes les ressources**
    - **Compose avec les conflits de demandes d'utilisation efficace et équitable des ressources**
  - **Le SE est un programme de contrôle**
    - **Contrôle l'exécutions des programmes (i.e. processus) pour prévenir des erreurs et la mauvaise utilisation de l'ordinateur.**

La définition du système d'exploitation

- **Alors, c'est quoi un système d'exploitation?**
- **Pas de définition universel acceptée**
- **"Tout ce que le fournisseur livre lors de la commande d'un SE" est une première approximation**
  - **Mais cela varie beaucoup.**
- **"Le programme en opération continu" est celui utilisé dans ce cours**
  - **Ceci est le noyau**
  - **Tout autre programme est un programme système (livré avec le SE) ou un programme d'application.**

Les programmes systèmes

- **Font ils partie du système d'exploitation**
- **Tout ce qui n'est pas dans le noyau, mais livré avec le SE**
  - **Tous dépend du SE et du fournisseur**
  - **Peut donner plusieurs services systèmes, i.e. les commandes UNIX (CLI) sont des programmes systèmes pour accomplir des tâches systèmes.**
  - **La perception des usagers du SE provient des programmes systèmes et non pas directement des appels systèmes au noyau.**

Programme systèmes

- **Les programmes systèmes (aussi appelés programmes utilitaires) fournissent un environnement de développement et d'exécution de programme. Les services sont:**
  - **Gestions de fichiers – i.e. copy, rm, ls, mkdir**
  - **Information d'état – i.e. ps, who, regedit**
  - **Modification de fichiers - éditions**
  - **Service de langage de programmation – i.e. cc, javac**
  - **Chargement et exécution de programmes – loaders, débuggeurs**
  - **Communications – ssh, ftp**
  - **Programmes d'application – fureteurs, pages électroniques, jeux**

Services offerts aux programmes d'utilisateurs

- **Opérations d'entrée/sortie**
  - **L'accès au matériel se fait par le noyau pour le programme exécutant**
- **Communications**
  - **Communication entre les différents programmes d'un même ordinateur ou avec ceux d'autres ordinateurs**
  - **Peut s'effectuer avec la mémoire partagée ou des messages**
- **Composer avec les erreurs**
  - **Détection**
    - **Erreurs du matériel (internes ou externes): la mémoire, la défaillance d'un dispositif E/S**
    - **Erreurs du logiciel: débordements, interdiction d'accès à une case mémoire**
    - **Impossibilité pour le SE de satisfaire une requête**
  - **Réaction: juste rapporter l'erreur à l'application, essayer de nouveau l'opération, suspendre l'application**

Services pour assurer l'efficacité et le bon fonctionnement

- Allocation et gestion des ressources
  - Nécessaire pour desservir plusieurs utilisateurs et plusieurs programmes
  - Certaines ressources ont leur code de gestion spécifique:
    - UCT, mémoire principale, système de fichier
  - D'autres sont gérés via un code général – E/S
- Comptabilité
  - Statistiques d'utilisation des ressources par les utilisateurs
- Protection et sécurité
  - Empêcher les intrus (usagers non autorisés) d'accéder au système
  - Empêcher les usagers d'accéder aux ressources qui ne leur sont pas destinées (Gestion de droits d'utilisation de ressources)

Interface pour utilisateurs - CLI

L'interface de ligne de commande (CLI - Command Line Interface) permet une entrée de commande directement du clavier

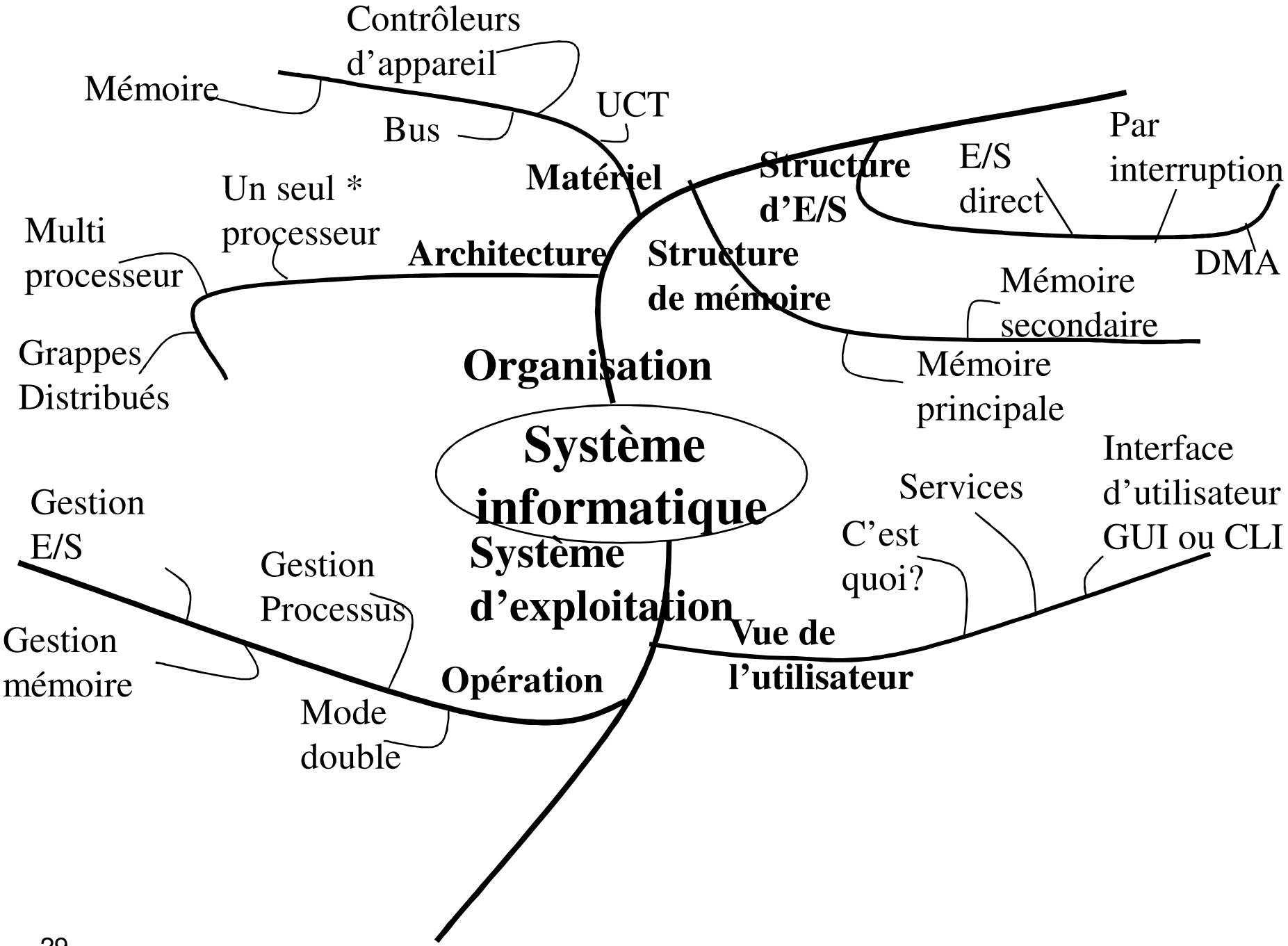
- **L'interpréteur de commande est un programme qui lit les commandes entrées par l'utilisateur**
  - **Souvent appelé le « shell » (terminologie UNIX)**
- **L'exécution d'une commande se fait d'une des deux façons suivantes:**
  - **L'interpréteur exécute la commande**
    - Des instructions de programmation permettent à l'interpréteur d'exécuter des programmes « shell »
  - **La command est utilisé pour démarrer un programme séparé (e.g. un programme système)**

Interface pour utilisateurs - GUI

- Interface conviviale qui représente une plateforme de travail
  - Avec souris, clavier, et moniteur
  - Icônes représentant les fichiers, programmes, actions, etc.
  - Inventé par Xerox PARC
- Beaucoup de systèmes comprennent des interfaces CLI et GUI.
  - Microsoft Windows est un GUI du type CLI “command” shell
  - Apple Mac OS X contient l’interface GUI “Aqua” et un noyau UNIX et donc le “shell”
  - Solaris et Linux sont du type CLI avec des interfaces GUI optionnelles (Java Desktop, KDE)
- Quelle interface préférez-vous?

Interfaces du SE

- CLI et GUI – interfaces pour l'usager
- Quelles sont les autres interfaces du système d'exploitation?
  - Interface pour programmes qui opèrent sur l'ordinateur et font des demandes de services au SE
    - L'interface d'appel de système
  - Interface avec le matériel
    - Interruptions, pilotes ⟷ contrôleurs d'appareil


Système informatique
Matériel
Contrôleurs d'appareil
Mémoire
Bus
UCT
Organisation
Architecture
Un seul *
processeur
Multi processeur
Grappes
Distribués
Structure de mémoire
Mémoire secondaire
Mémoire principale
Structure d'E/S
E/S direct
Par interruption
DMA
Système d'exploitation
Opération
Mode double
Gestion Processus
Gestion mémoire
Gestion E/S
Vue de l'utilisateur
C'est quoi?
Services
Interface d'utilisateur GUI ou CLI

Opérations du système d'exploitation

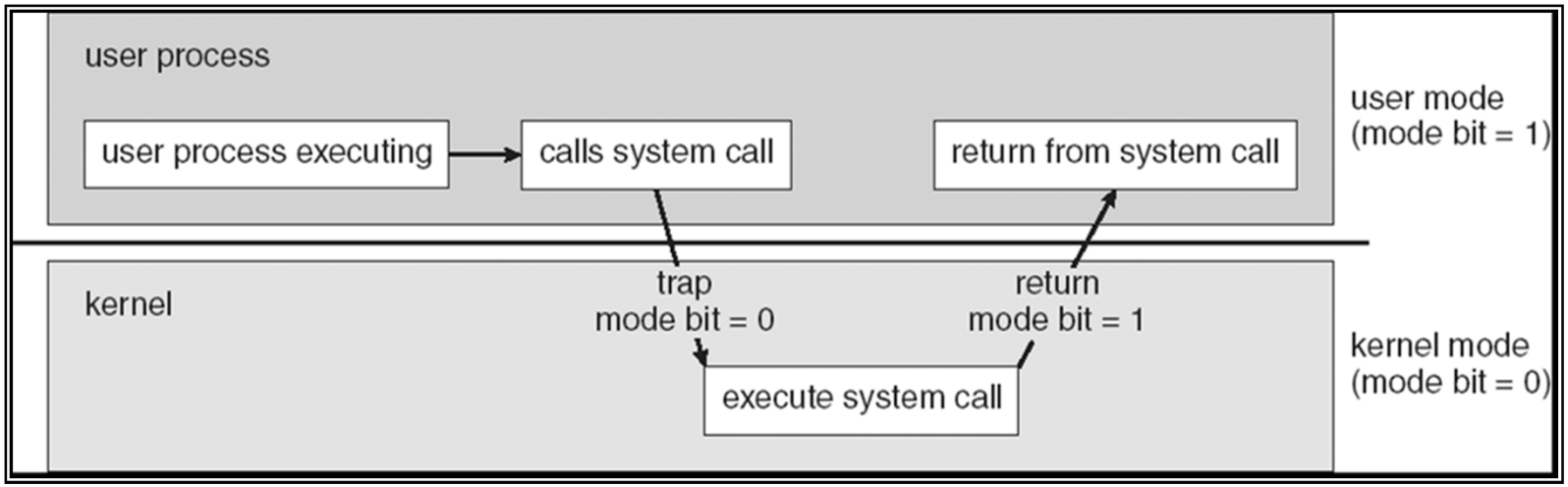
- Le SE opère à l'aide d'interruptions
- Les interruptions proviennent du matériel ET du logiciel
  - Cliquage de souris, division par zéro, demande de service du SE
  - Interruption de minuterie (temps de processus fini), erreur d'accès à la mémoire (processus veut en modifier un autre ou le SE).
- Certaines opérations devront seulement se faire par un programme fiable.
  - Accéder le matériel et registres de gestion de la mémoire.
  - Un programme maléfique d'utilisateur pourrait endommager d'autres processus, s'accaparer illégalement le système, …
  - Solution: opération en mode double.

Le mode usager et le mode noyau

- **L'opération en mode dual permet au ES de se protéger ainsi que d'autres composantes**
  - **Mode usager et mode noyau (ou supervision)**
  - **Le bit mode est fournit par le matériel**
    - **Distingue le mode (usager ou noyau)**
    - **Certaines instructions sont exécutables uniquement dans le mode usager**
    - **Un appel de système permet de changer au mode noyau, un retour d'appel le rechange au mode usager.**

Transition du mode usager au mode noyau

- **Une minuterie empêche les processus de s'accaparer du système**
  - **Correspond à une interruption après un délai de temps**
  - **Le SE décrémente un compteur**
  - **Lorsque le compteur est à zéro, un changement ou terminaison de processus se produit.**
  - **Configurer avant de donner le contrôle à un processus pour pouvoir le reprendre ou terminé le programme.**



L’appel système

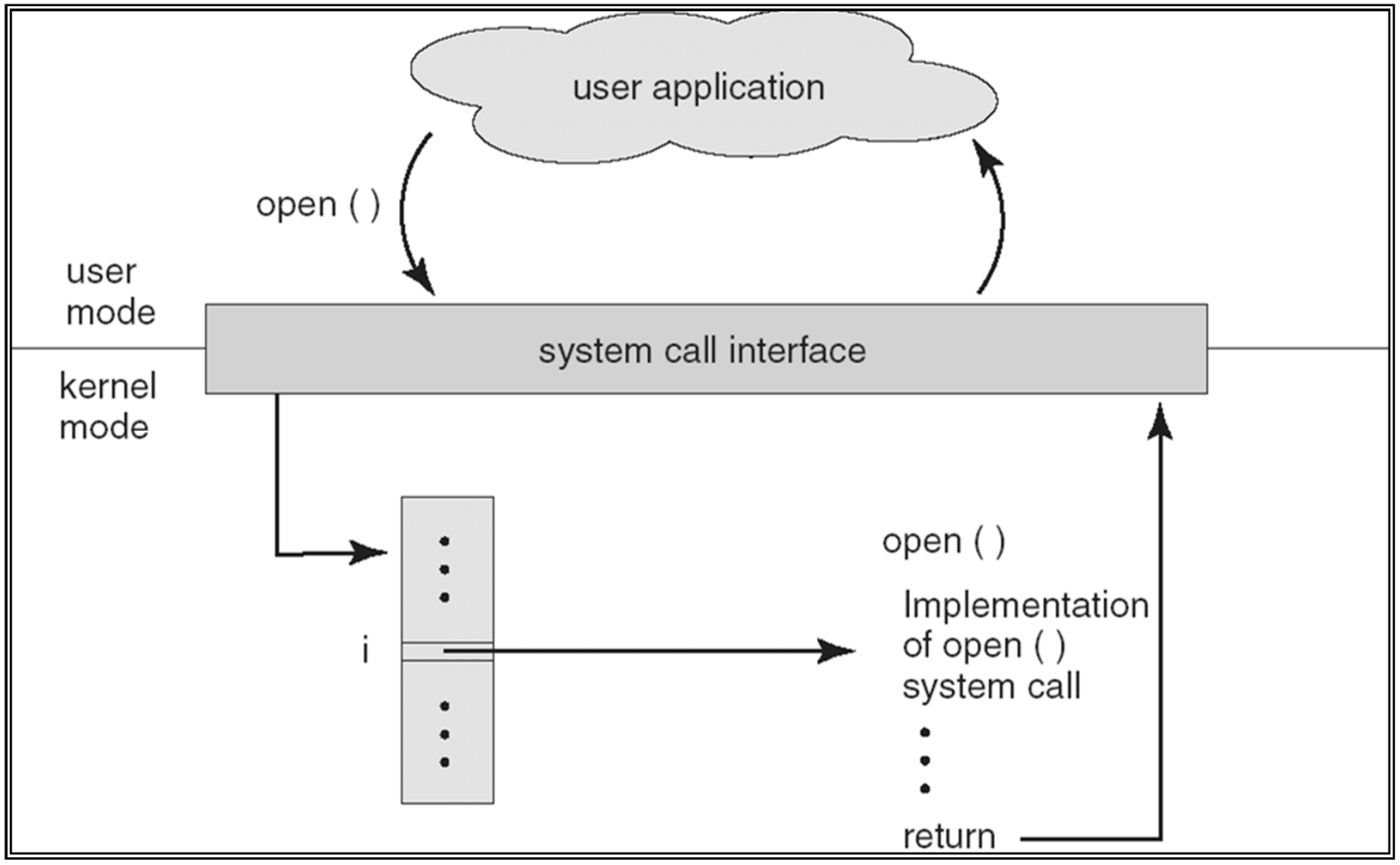
- L’interface qui offre les services du SE au programmes
  - Le control de processus:
    - pour exécuter un programme.
  - La gestion de fichier:
    - création/ouverture/lecture/écriture d’un fichier, listage d’un répertoire.
  - La gestion d’appareil:
    - Demande d’accès/libération d’appareils
  - La gestion d’information:
    - gestion du temps, des attributs de processus et de fichiers
  - La communication:
    - Ouverture/fermeture de connexion, expédition/réception de messages

Appel système (suite)

- **Normalement écrit dans un langage de programmation de haut niveau (par exemple le C).**
- **Implémenter avec une interruption logicielle**
  - **L'interruption logicielle change le bit mode au mode noyau et fait appel au sous-programme approprié selon un tableau d'appels systèmes et le numéro d'interruption**
    - **Exemple Linux:**

    http://docs.cs.up.ac.za/programming/asm/derick_tut/syscalls.html
- **À la fin du sous-programme, le mode revient au mode usager et des valeurs sont retournées au programme évocateur.**

API de l'appel système

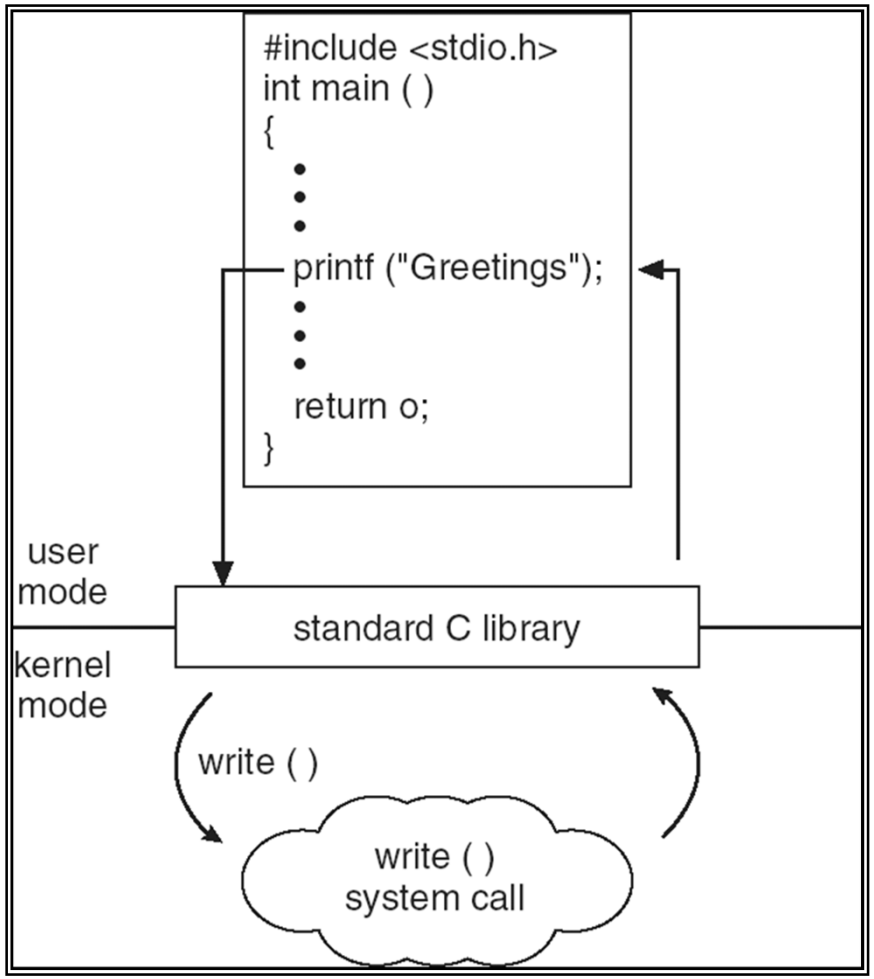


Comment accéder aux appels systèmes

- Accéder le plus souvent par des programmes à l'aide d'une interface de programmation d'application (API) et non pas directement par appel de systèmes
- APIs commun:
  - Win32 pour Windows
  - API POSIX pour systèmes POSIX
    - UNIX, Linux et MAC OS X
  - API Java pour la machine virtuelle Java (JVM)
- Le programme appelant ignore tout de l'implémentation de l'appel de système.
  - Obéit tout simplement au normes de l'API: les paramètres à fournir, les valeurs de retour, et l'opération désirée
  - Les détails de l'interface du SE sont cachés derrière l'API.
    - Géré par la librairie de l'API (ensemble de fonctions fournit avec le compilateur)
- Possible d'utiliser les appels système directement

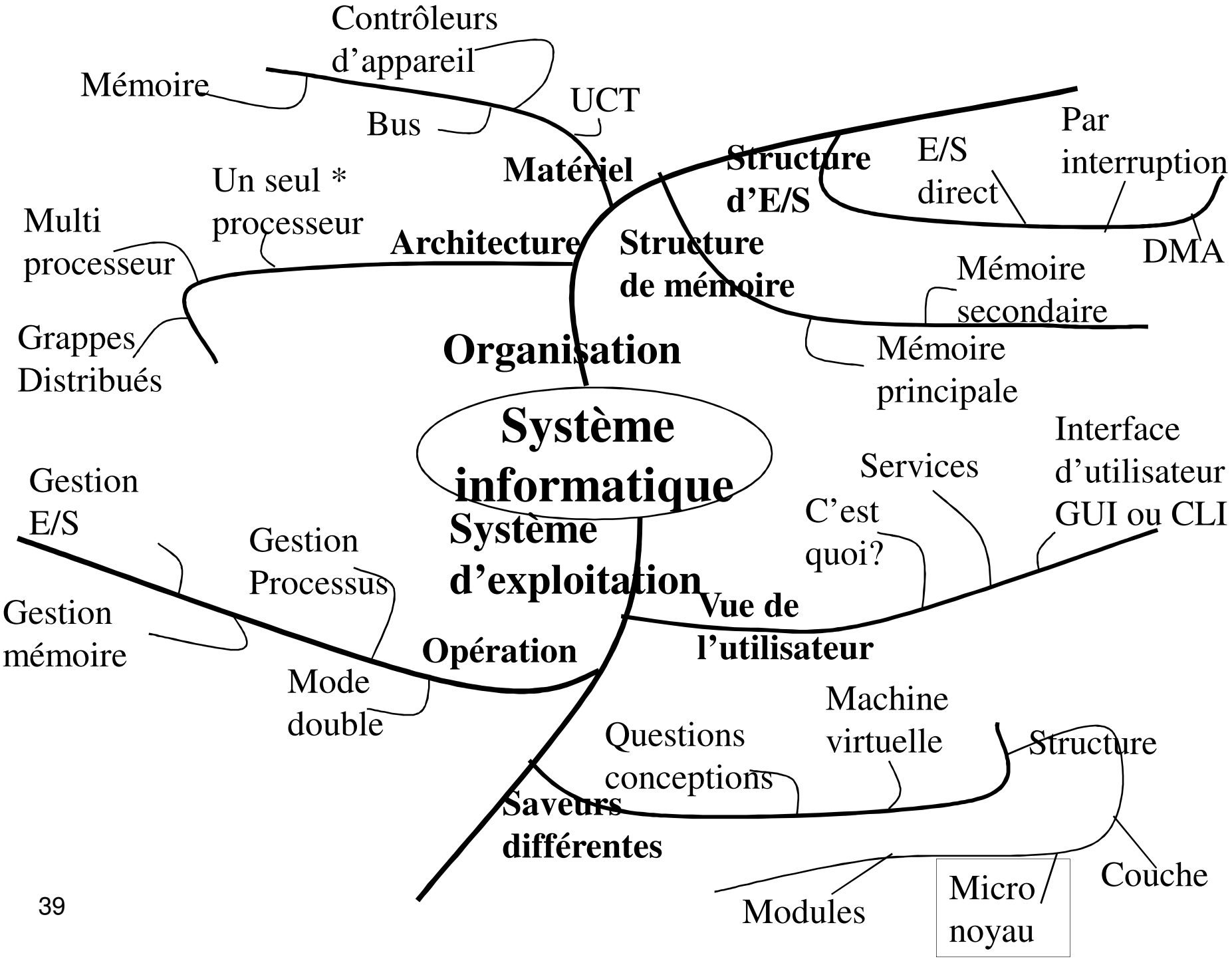
Exemple de la librairie standard de C

- **La fonction printf() qui fait un appel de système write()**



Opérations principales du système d'exploitation (SE)

- Gestion de processus
  - Un processus avec opération comprend un seul compteur de programme
  - Le SE gère les ressources requises par les processus
    - UCT, mémoire, E/S, fichiers
    - Données d'initialisation
  - Le SE gères les activités des processus: création, destruction, interactions entre processus, etc.
- Gestion de la mémoire
  - La gestion de mémoire détermine quel processus est actif et à quel moment il occupe la mémoire afin d'optimiser l'utilisation de l'UCT et la réponse de l'ordinateur aux utilisateurs
- Gestion de la mémoire secondaire
  - Le SE donne une vue uniforme et logique de l'information stocké en mémoire secondaire
  - Système de fichier, mémoire de masse
- Le sous-système E/S
  - Un des rôles du SE est de cacher les différentes particularités des appareils de l'utilisateur


Système informatique
Matériel
Contrôleurs d'appareil
Mémoire
Bus
UCT
Architecture
Un seul *
processeur
Multi processeur
Grappes
Distribués
Organisation
Structure de mémoire
Structure d'E/S
E/S direct
Par interruption
DMA
Mémoire secondaire
Mémoire principale
Système d'exploitation
Opération
Gestion E/S
Gestion Processus
Gestion mémoire
Mode double
Vue de l'utilisateur
C'est quoi?
Services
Interface d'utilisateur GUI ou CLI
Saveurs différentes
Questions conceptions
Machine virtuelle
Structure
Modules
Micro noyau
Couche

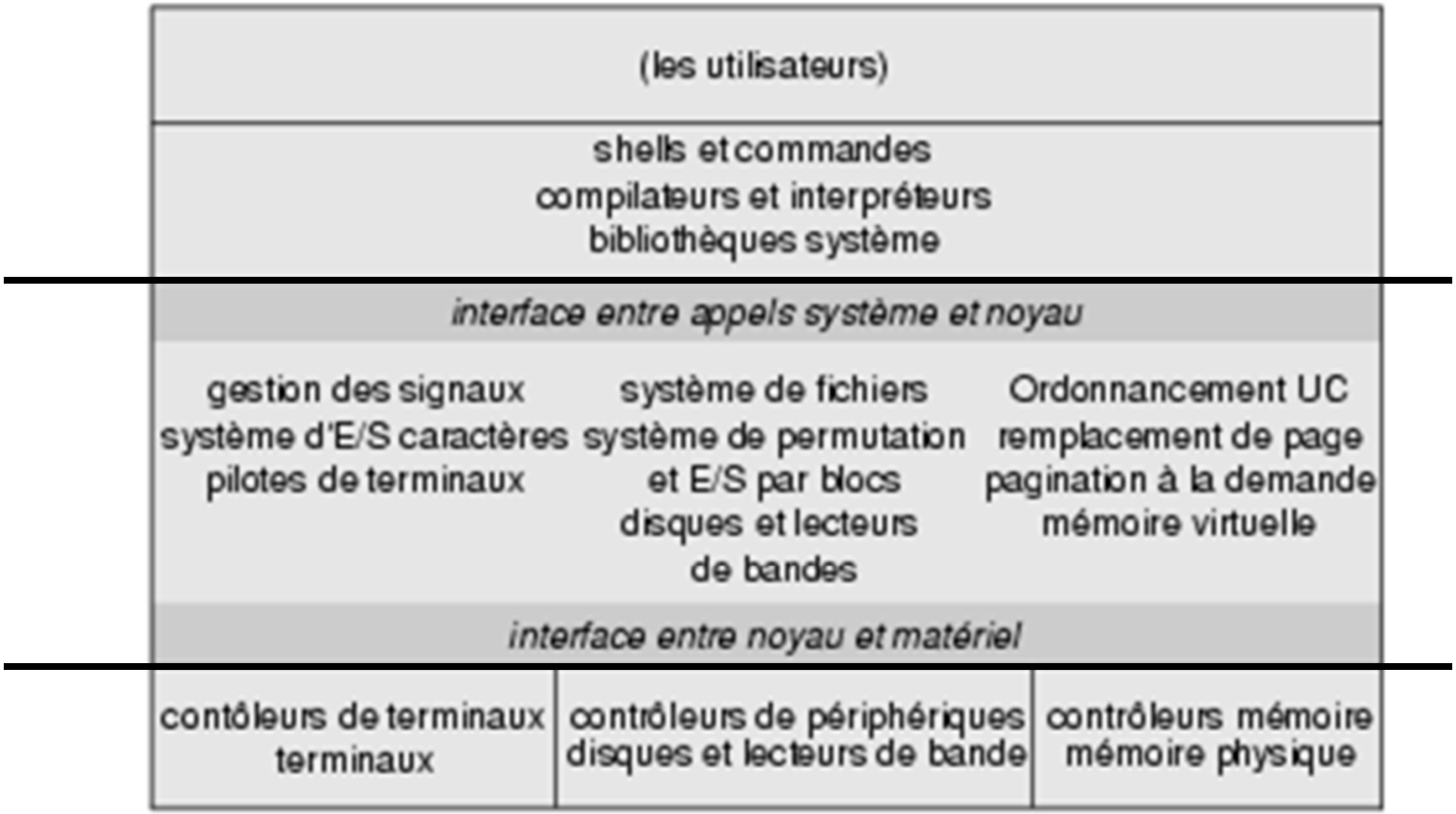
Conception et réalisation des SEs

- La conception du SE est principalement affecté par le choix de matériel et du type de système
  - En gros, temps-partagé, un utilisateur, multi-utilisateurs, distribués, temps-réel, usage général
- Besoins d'utilisateurs versus besoins des système
  - Besoins d'utilisateurs – facile à utilisé, facile à apprendre, fiable, et rapide
  - Besoins de système – facile à concevoir, simple à réaliser et à entretenir, ainsi que flexible, fiable, sans erreur, et efficace.
- Implémentation
  - Traditionnellement en assembleur
  - Aujourd'hui, surtout en C, avec des petites sections en assembleur (pilotes, manipulation de registres)

Structure du système

- Structure interne des SEs varient
  - Puisque les besoins varient
- Matériel simple, fonctions simples
  - Structure monolithique simple
- Plus de resources et fonctions complexes
  - Structure de courches
  - MS-DOS et UNIX traditionnel sont des SEs monolithiques qui utilsent une structure de couches.
- Encore plus de ressources et fonctions, avec une concentration sur une conception flexible et élégante
  - Structure de micronoyau (micro-kernel) (MACH, QNX, Windows NT)
- Flexibilité et efficacité
  - Structure modulaire (Solaris, Windows NT)

Structure UNIX: peu de couches



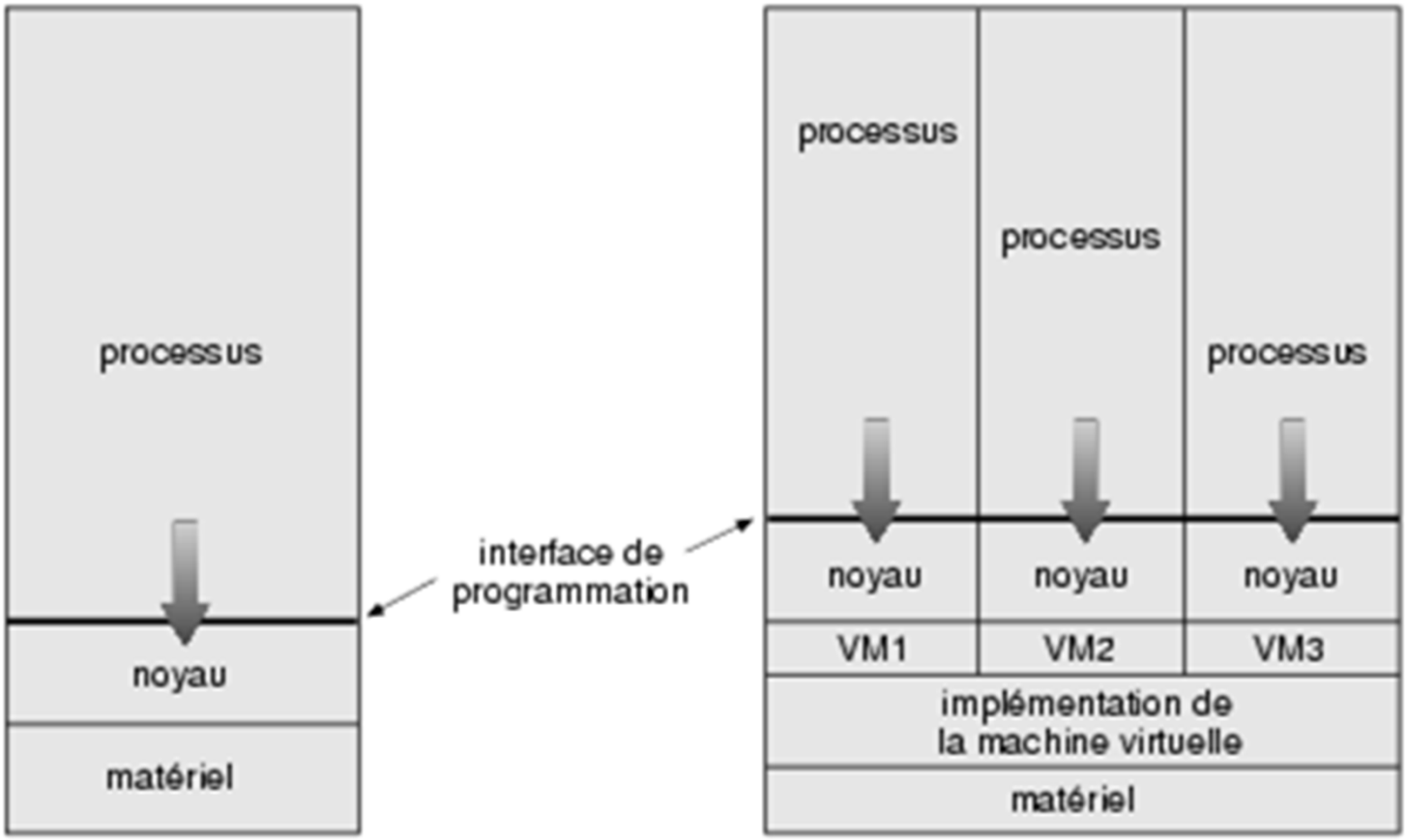
Machines virtuelles: le problème et la solution

- **Comment permettre d'opèrer différents SE sur une seule machine physique?**
- **Pas évident, car chaque SE demande accès direct au matériel**
- **SOLUTION: Un programme qui crée une couche qui est mise à la disposition de plusieurs machines physiques *virtuelles***
- **Chaque couche peut utiliser un SE différent**

Modèle de système

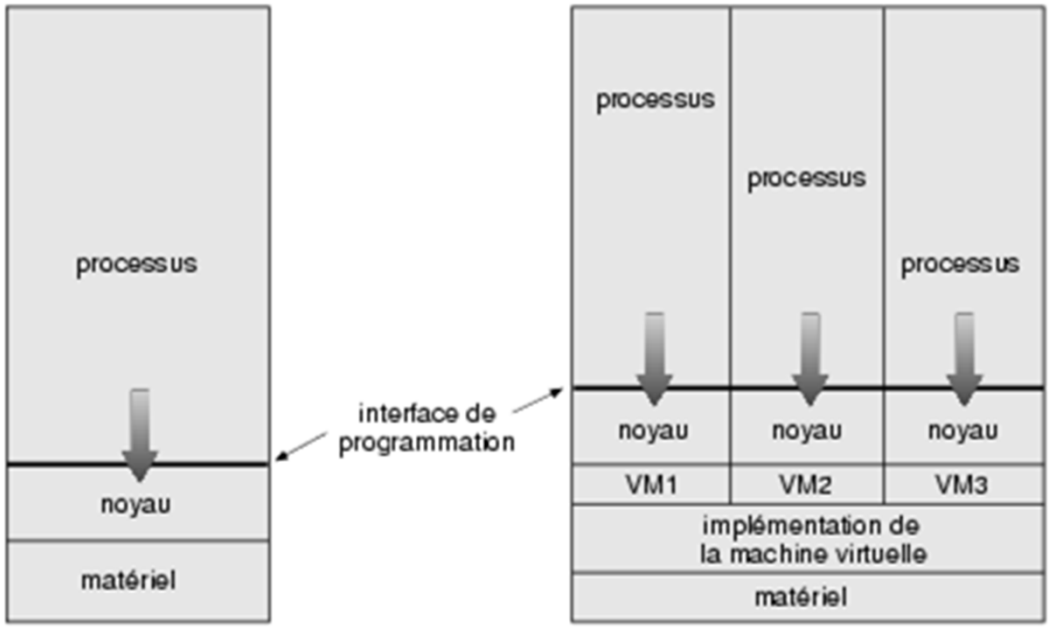
(a) Une seule machine réelle et un seul noyau

(b) plusieurs. machines virtuelles et plusieurs noyaux



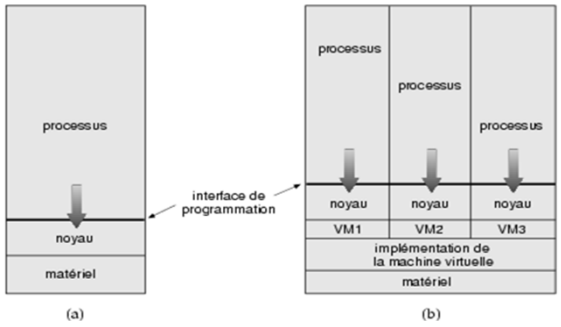
Fonctionnement

- **Le système VM laisse exécuter normalement les instructions non privilégiées**
- **Les appels au système sont exécutés par le système VM et les résultats sont passés à la machine virtuelle où le processus est exécuté**



Avantages

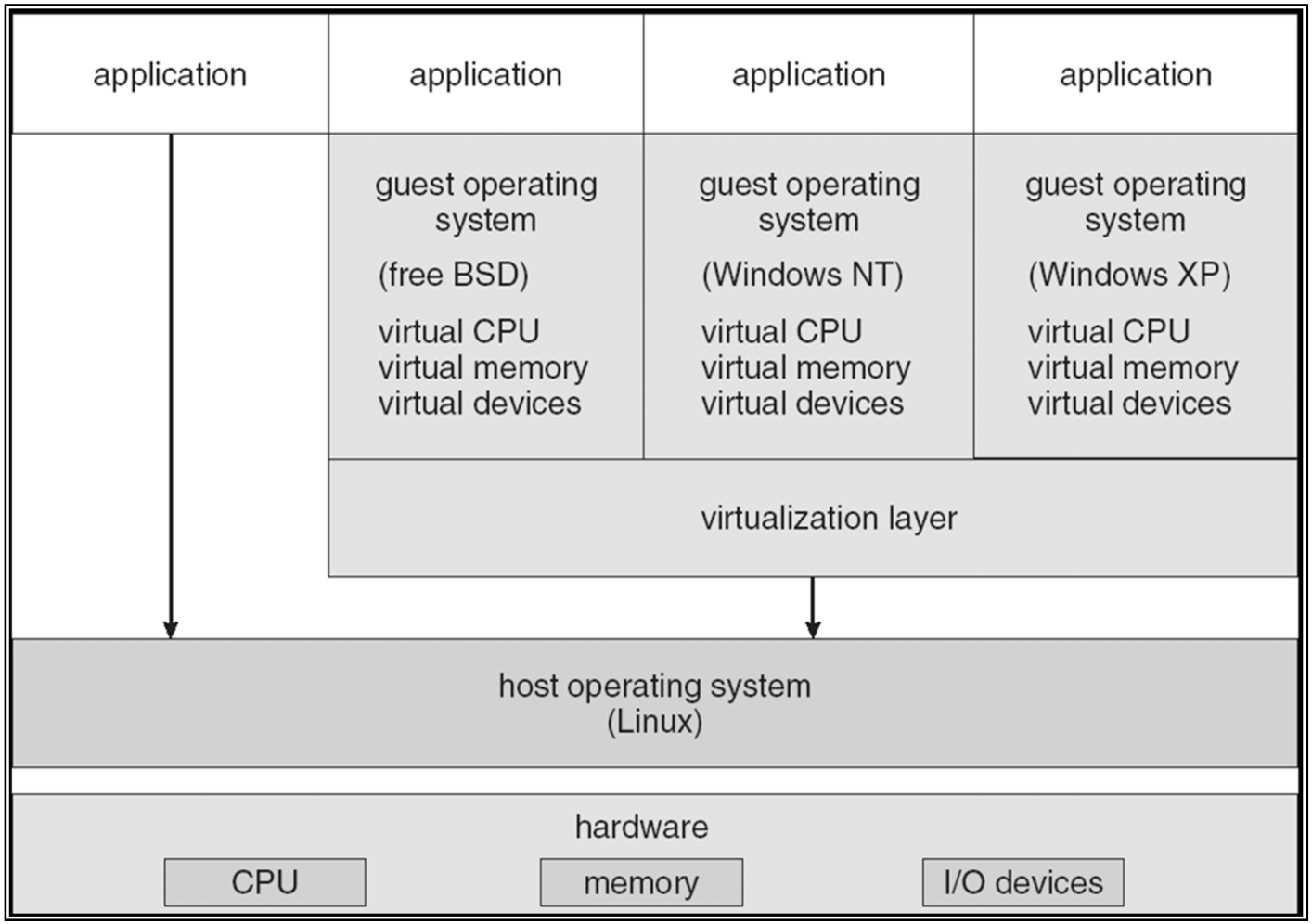
- **Chaque machine virtuelle peut utiliser un SE différent!**
- **En théorie, on peut bâtir des machines virtuelles sur des machines virtuelles!**
- **Protection complète, car les machines virtuelles sont complètement isolées les unes des autres**
- **Un nouveau SE peut être développé sur une machine virtuelle sans déranger les autres**



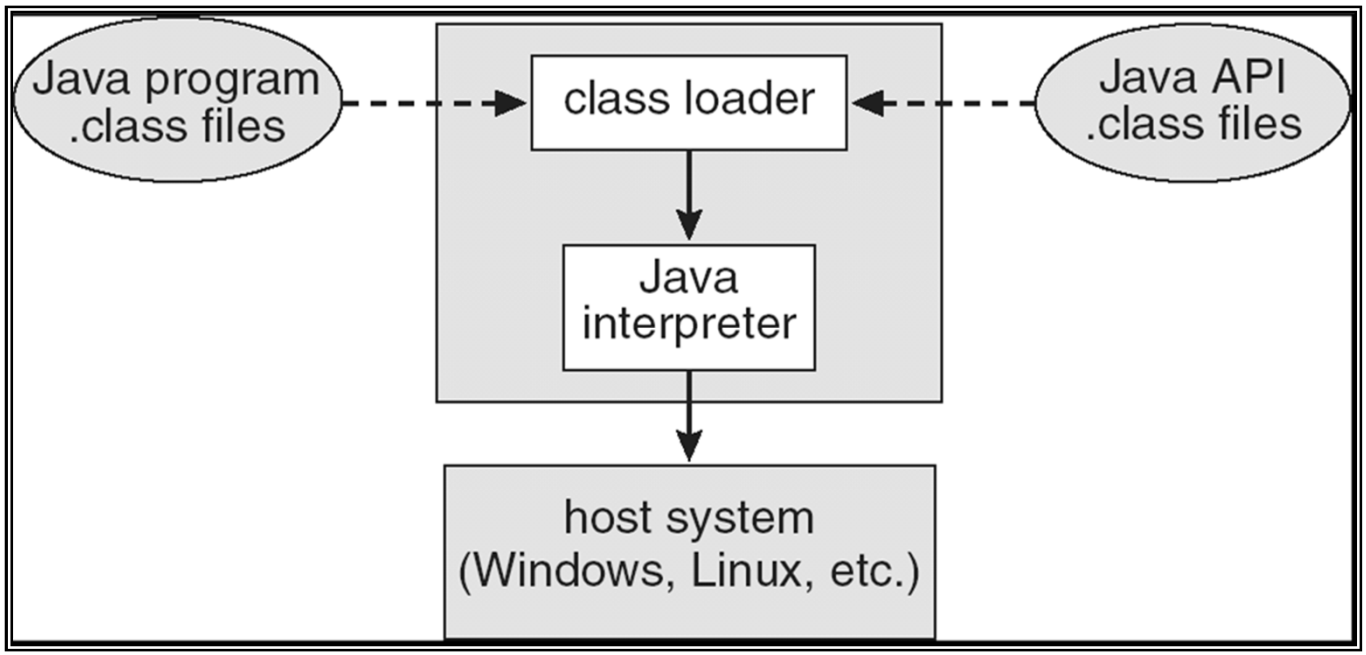
Implémentations

- **Le concept de VM est très utilisé pour permettre d'opérer un SE sur un autre**
- **P.ex. SUN, Apple, Linux permettent d'opérer Windows sur leur plateforme,**
- **Ils doivent fournir à Windows un environnement que Windows reconnaît comme un environnement Intel habituel**

VMWARE sur Linux

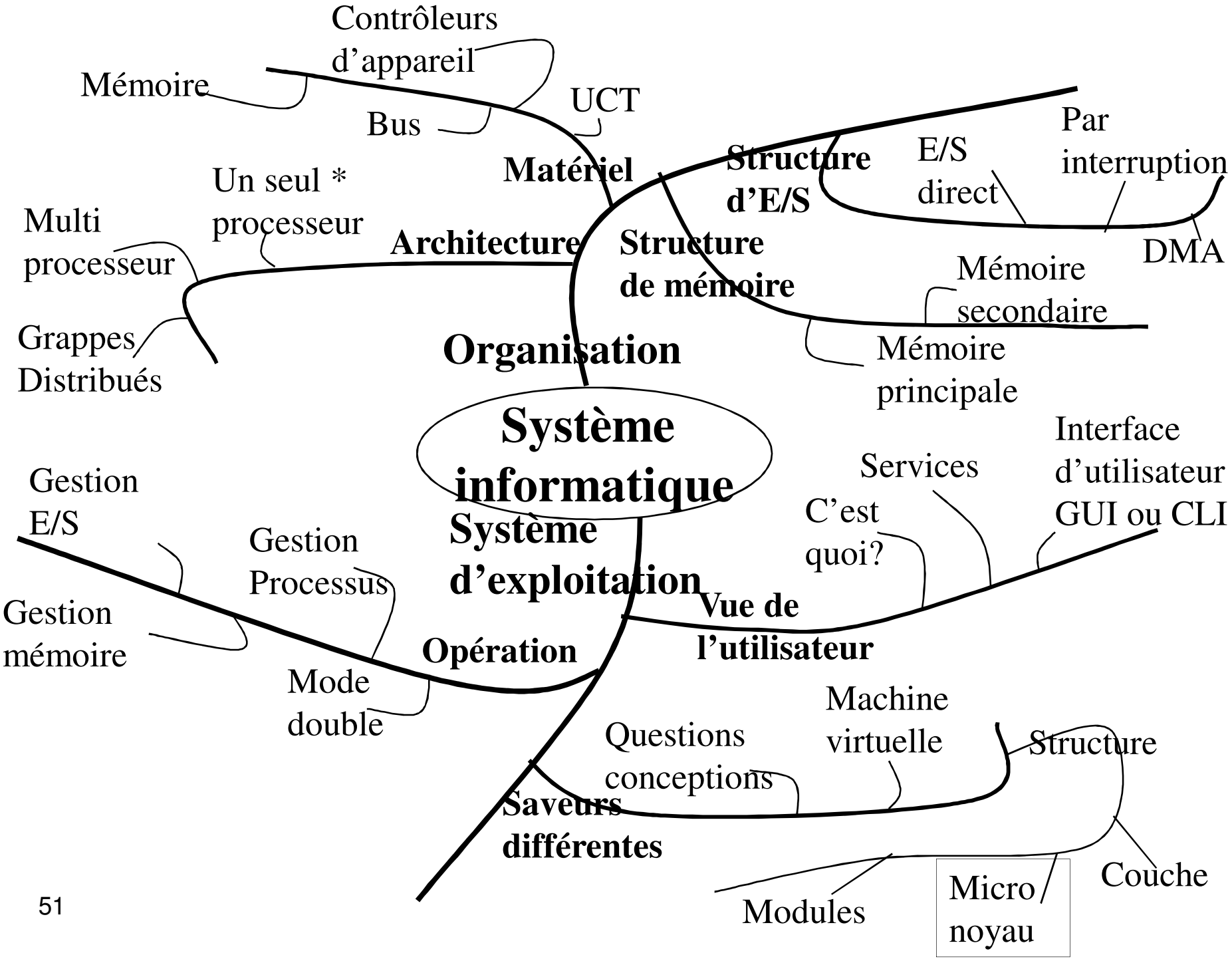


Machine virtuelle de Java



Ce que nous n'allons pas étudier

- Système distribués
- Systèmes en temps réel imbriqué
- Systèmes multimédia
- Ordinateurs de poche
- Poste à poste (peer to peer)
- Système d'exploitation WEB


Système informatique
Matériel
Contrôleurs d'appareil
Mémoire
Bus
UCT
Architecture
Un seul *
processeur
Multi processeur
Grappes
Distribués
Organisation
Structure d'E/S
E/S direct
Par interruption
DMA
Structure de mémoire
Mémoire secondaire
Mémoire principale
Système d'exploitation
Opération
Gestion E/S
Gestion Processus
Gestion mémoire
Mode double
Vue de l'utilisateur
C'est quoi?
Services
Interface d'utilisateur GUI ou CLI
Saveurs différentes
Questions conceptions
Machine virtuelle
Structure
Modules
Micro noyau
Couche