

# Les Systèmes d'Exploitation: Concepts et Programmation

Samia Bouzefrane

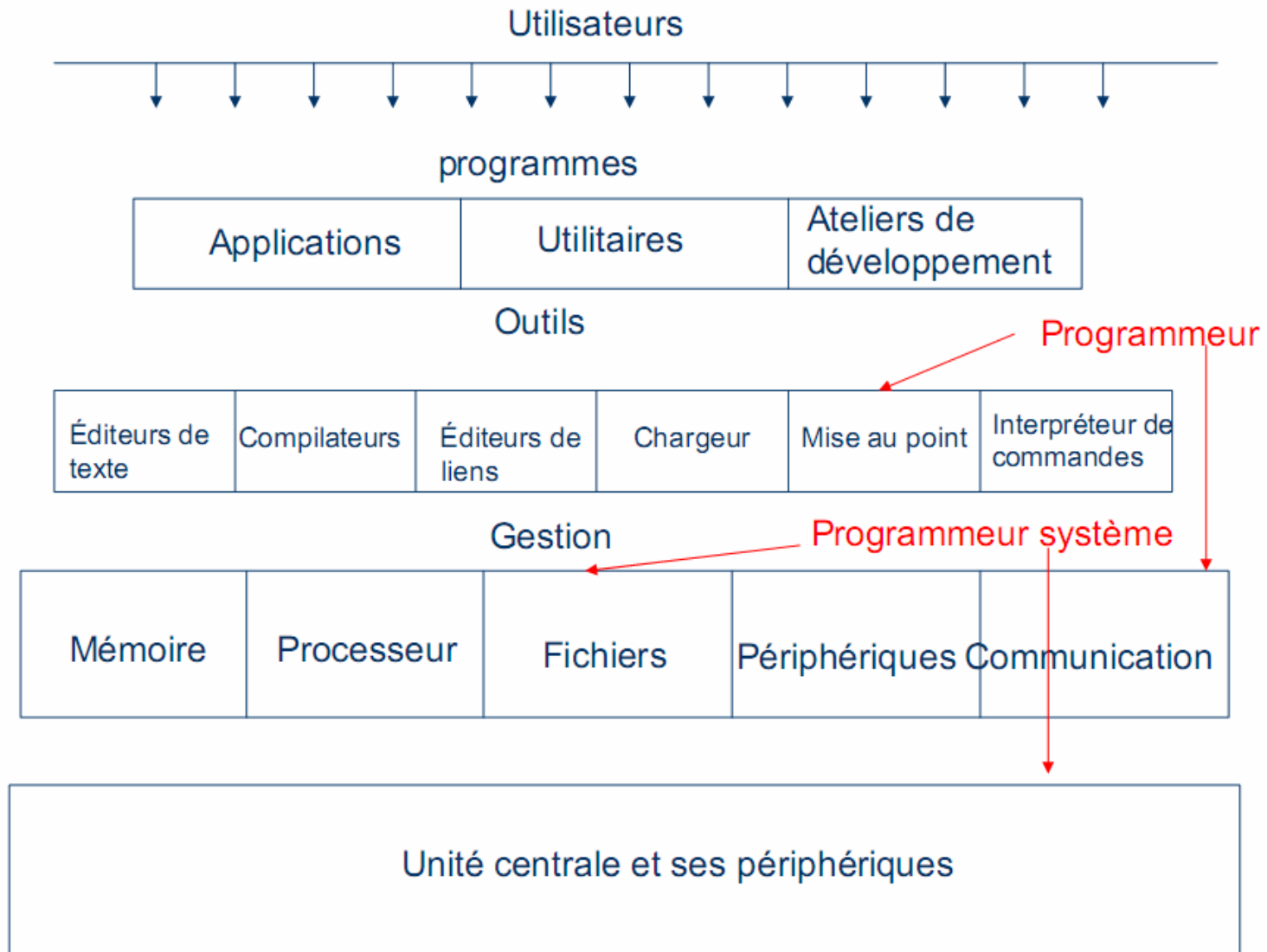
MCF en Informatique, CNAM

`samia.bouzefrane@cnam.fr`

`http://cedric.cnam.fr/~bouzefra`

# Introduction

# Structure générale d'un système informatique



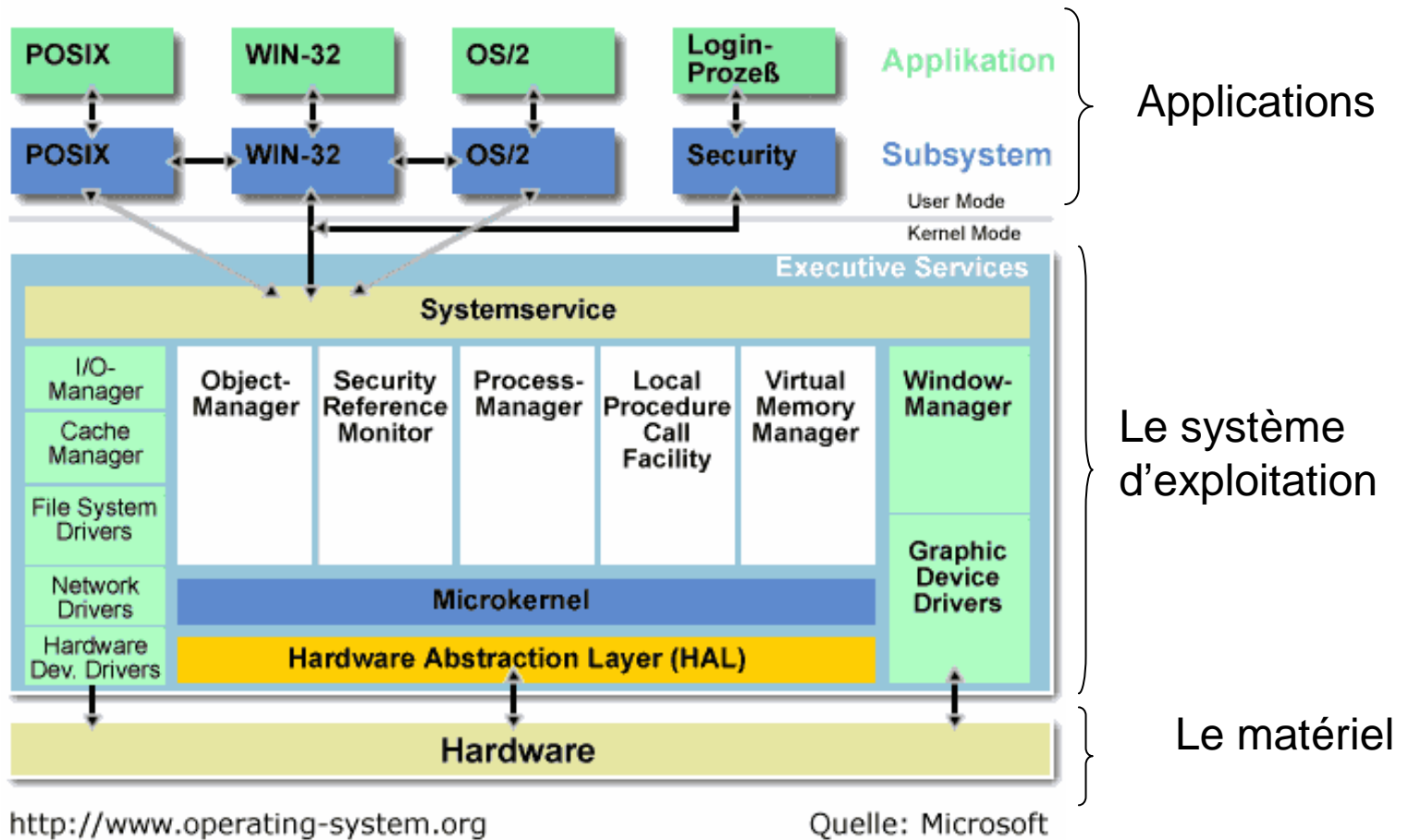
## Définition d'un système informatique

### ➤ *Un système informatique*

- Matériel
  - Unité centrale, disques, réseau, périphériques
- Système d'exploitation
  - Gestion utilisateurs, SGF, ressources (accès, partage, distribution, échange...)
- Programmes d'application
- Utilisateurs
  - Login, droits, accès

# Exemple : Architecture de Windows NT 4.0

Windows NT 4.0- Architektur



# Évolution des systèmes d'exploitation

- Batch
- Multiprogrammé
- Multi-utilisateurs centralisé
  - Temps partagé, transactionnel
- Distribué
- Temps réel

## Éléments d'un S.E.

**Matériel**

**Services de l'OS**

**Abstraction utilisateur**

Processeur	Gestion de processus, Ordonnancement, signaux, Protection, Synchronisation, Allocation	Processus
Mémoire	Allocation, protection, Mémoire virtuelle	Espace d'adressage
E/S	Gestion d'I/O ; DMA ; Synchronisation	Dispositif physique d'E/S Appels système
Disque	Gestion de la persistance des données ; SGF	Fichiers
Réseau	Protocole de sécurité ; SGF distribué	RPC ; Serveur de fichiers

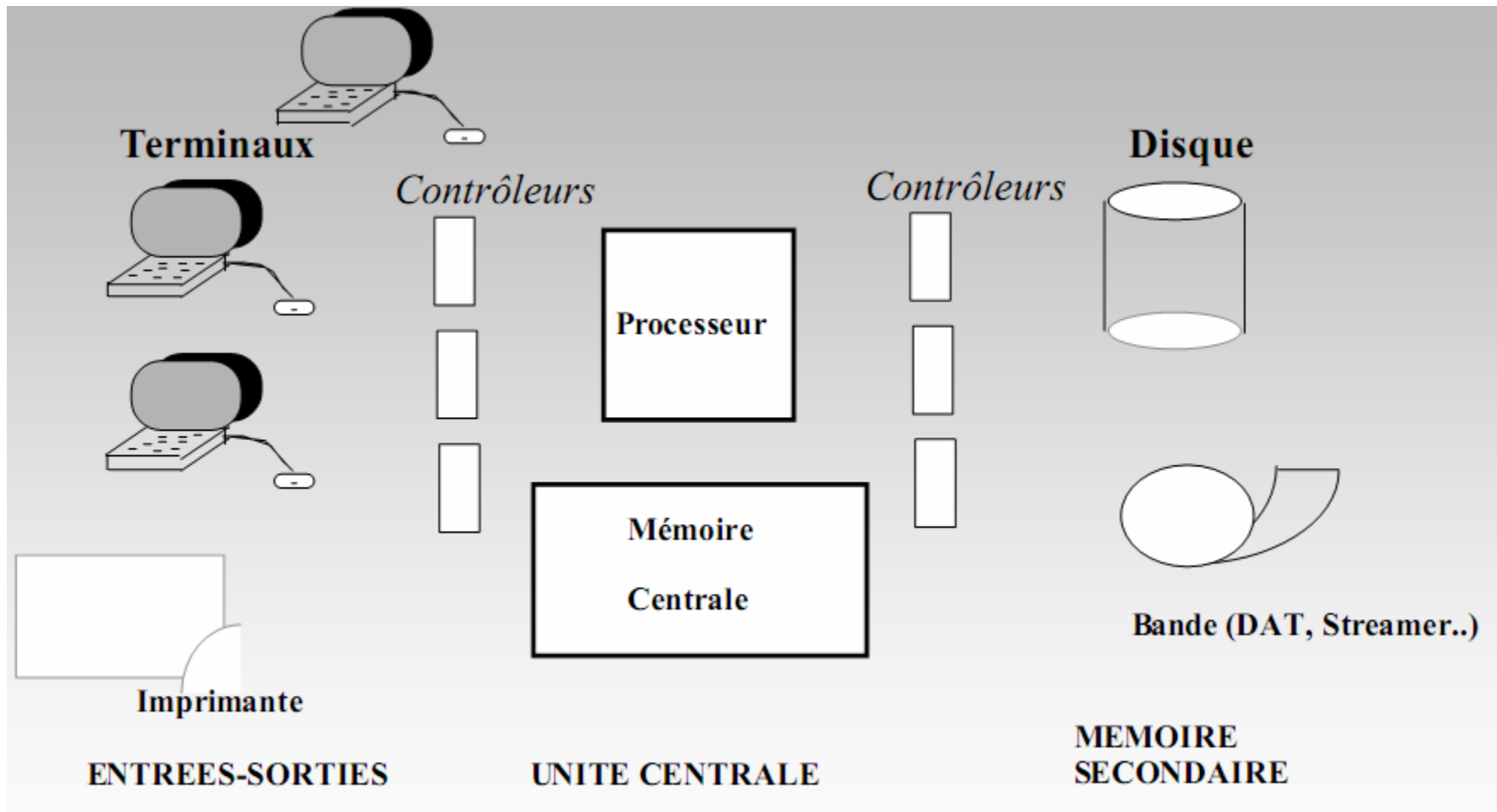
# Matériel



## Le matériel

- Principes de fonctionnement de l'UC
  - Calcul, contrôle, bus, pipe-line
- La hiérarchie mémoire
  - Registres, caches, MC, disques, réseau
- Les E/S
  - Interruptions, DMA

# Les constituants d'un ordinateur

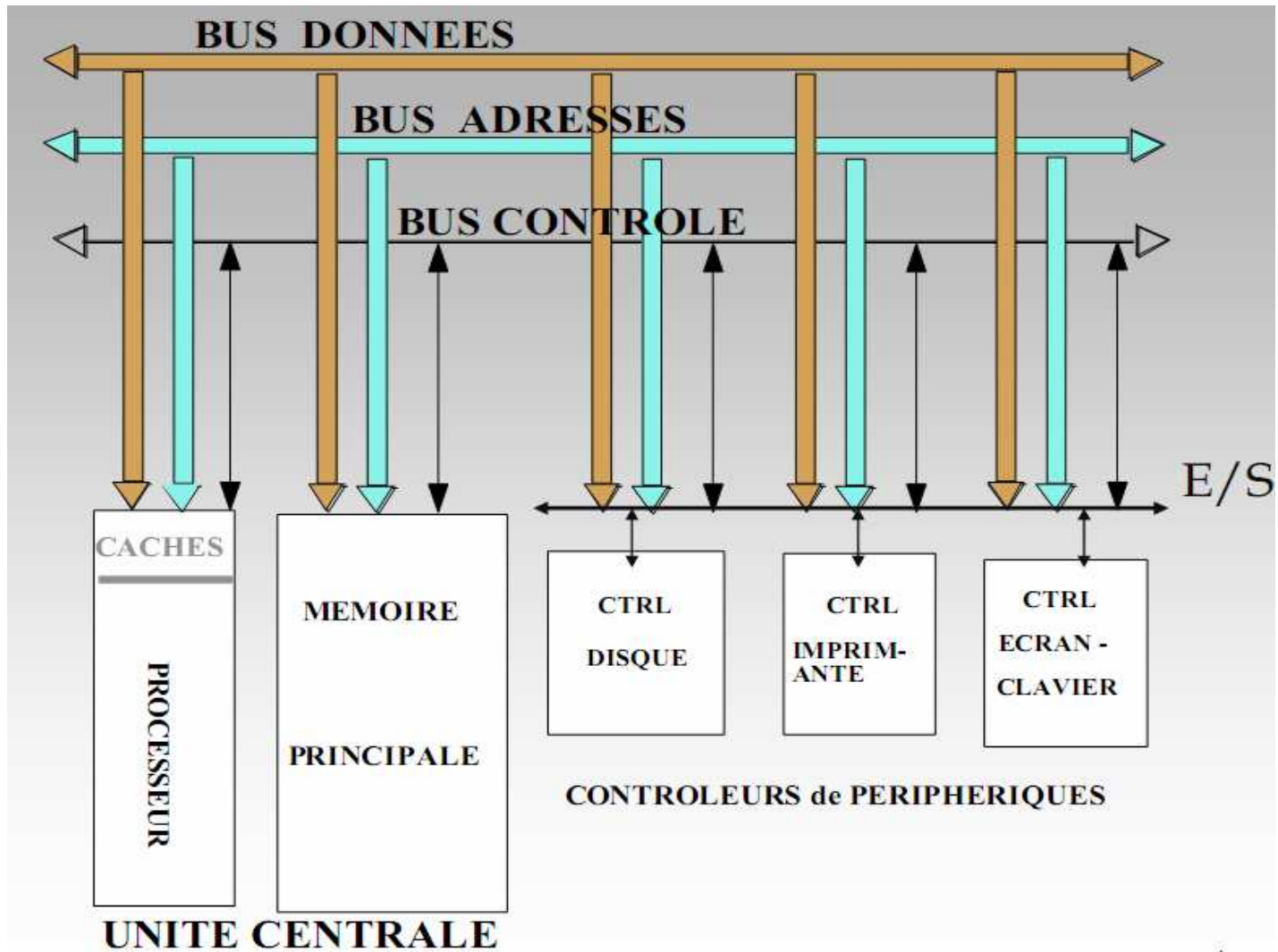


## Le processeur et la mémoire

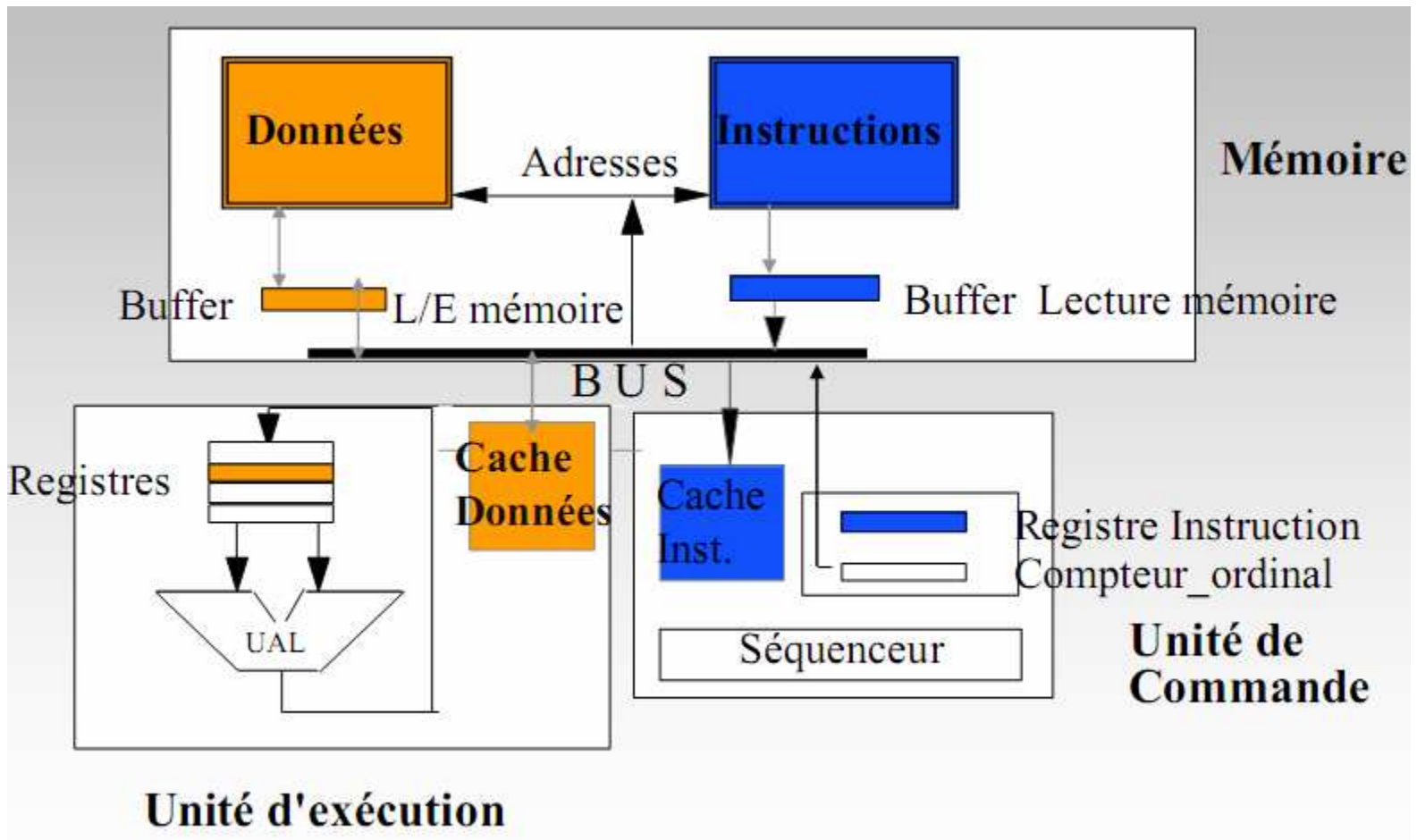
### ➤ *Caractéristiques principales*

- La largeur du bus données (64 bits) et la largeur du bus adresse (32/64bits)
- Le nombre de mots de la mémoire
- Les temps de cycle (processeur, bus mémoire)
- Le nombre de cycles par instruction
- Le débit d'instructions
- Le processeur est un chip VLSI ~ 100 millions de transistors
- La mémoire centrale est réalisée à partir de boîtiers de 16, 64, 256, 1024 Méga bits.

# Les bus de données/adresse/contrôle



# Le principe de Von Neuman



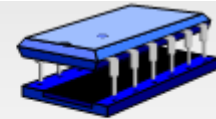
## Exemples de processeurs

### ➤ *Pentium IV 2002*

- 64 bits données et 32 bits adresses
- H = jusqu'à 2.53 Ghz.
- Cache L1 :12K/8K - L2 : 512 K on chip
- Nb transistors = 42 millions

### ➤ *IBM Power IV*

- 64 bits adresses et données
- H = 1.3 Ghz
- Cache 64 K/64K - L2 : 1,4 M. on chip 128 M off chip
- Nb transistors = 170 millions
- 2 procs + cache L2 on chip

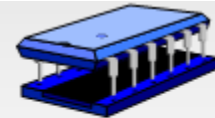


**PROCESSEUR**

## Exemples de processeurs

### ➤ *Pentium4 2004*

- 64 bits adresses et 32/64 données
- H = 3.6 Ghz
- Cache L1 :12K/16K - L2 : 1M on chip
- Nb transistors = 100 millions

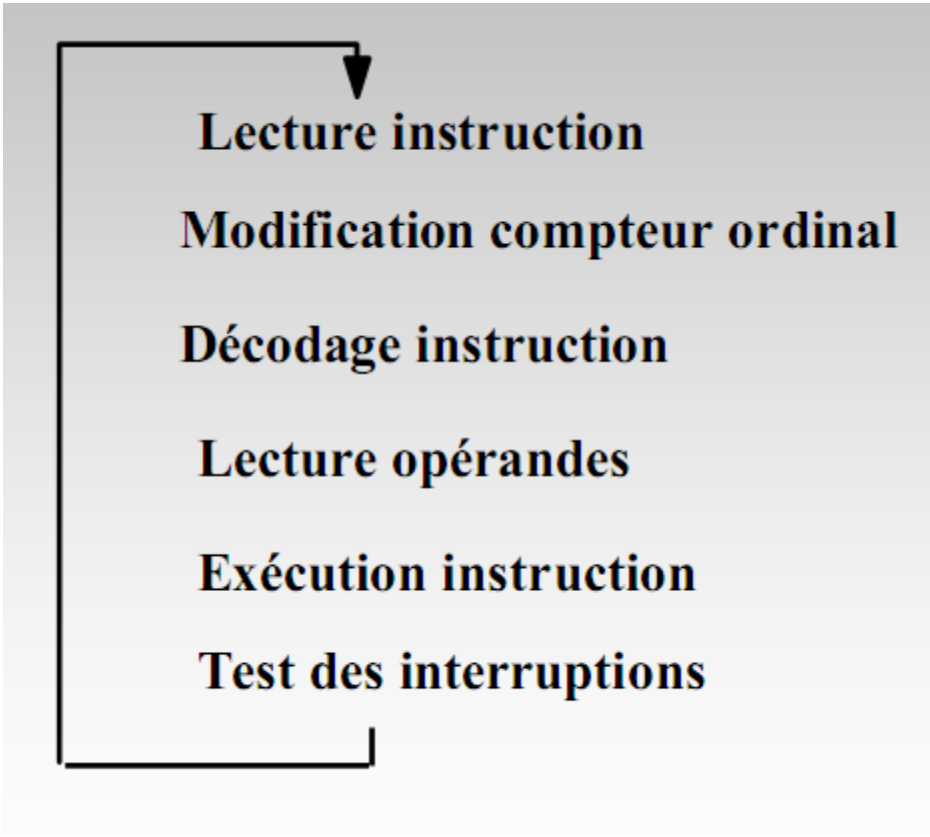


PROCESSEUR

### ➤ *IBM Power5 (Bi-processeur)*

- 64 bits adresses et données
  - H = 1.9 Ghz
  - Cache 64 K/32K - L2 : 1,9 M. on chip 36 M off chip
- 2 procs + cache L2 on chip

## **Algorithme exécuté par le processeur**



The diagram shows a vertical sequence of six steps within a rectangular frame. An arrow at the top points down to the first step. A line at the bottom of the frame connects to the top, forming a loop.

**Lecture instruction**

**Modification compteur ordinal**

**Décodage instruction**

**Lecture opérandes**

**Exécution instruction**

**Test des interruptions**

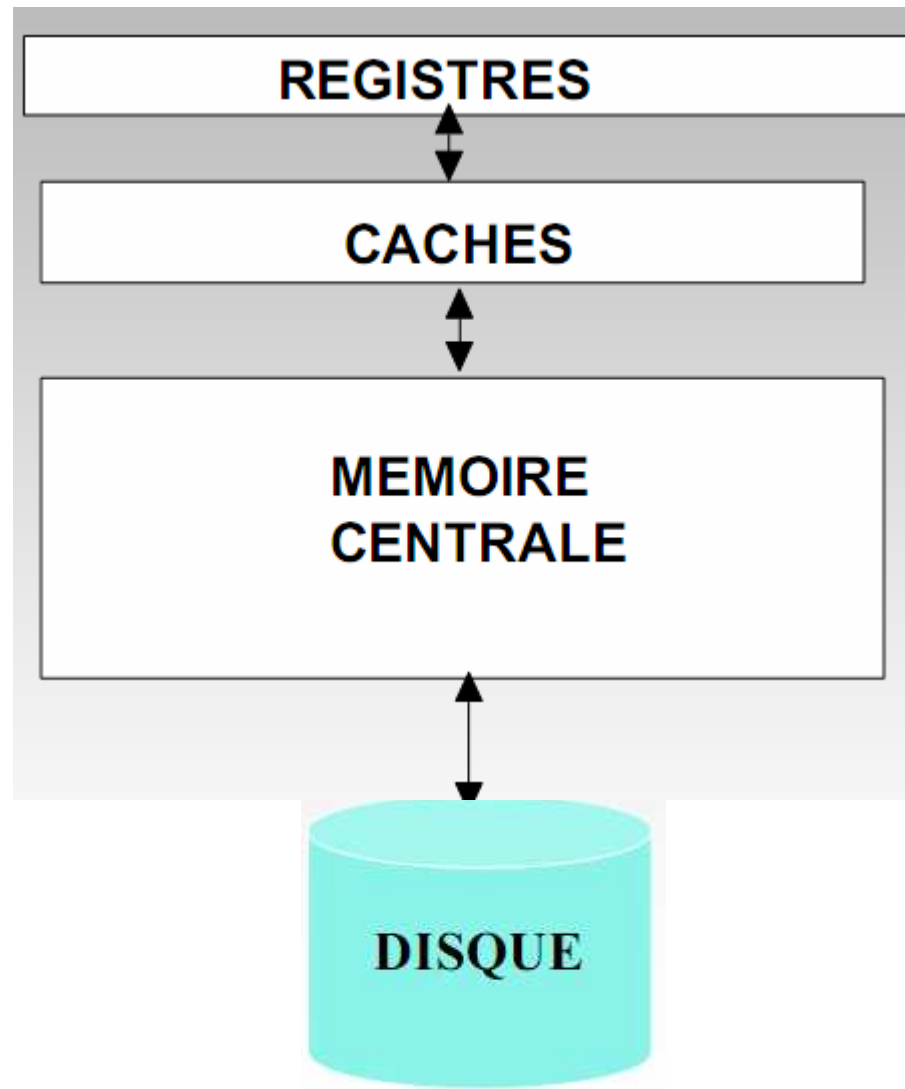


# La mémoire

## La hiérarchie mémoire

Gérée par :

- Le compilateur
- Un dispositif câblé
- Le système d'exploitation
- La création de fichiers (SGF)



## La mémoire centrale

- La mémoire, contenant des instructions et des données, est organisée en un ensemble de mots numérotés consécutivement à partir de 0
- Le N° associé à chacun de ces mots est l'adresse physique
- L'ensemble de ces adresses forme l'espace d'adressage physique de la mémoire

## Gestion de la mémoire centrale

- Taille mémoire limitée même si de plus en plus importante
- Découpage de la mémoire en :
  - zones de taille variable
  - pages de taille fixe : la *pagination*
  - segments (chaque segment contient un nombre fixe de pages mémoire) : la *segmentation*

# Les Entrées/Sorties

# Gestion des E/S et des fichiers

- Les dispositifs d'E/S
- Intégration des E/S dans le SE
- Le Système de Gestion de fichiers
  - Le type fichier
  - Les méthodes d'accès
  - L'allocation sur le disque
  - Le RAID
- Exemples UNIX, Linux, Windows

# Introduction

- Un écran vidéo ; un clavier
- Un disque magnétique
- Une imprimante

## *Caractéristiques communes :*

-Les périphériques d'E/S sont beaucoup plus lents que l'unité de traitement

→ Libérer l'UC des opérations d'E/S

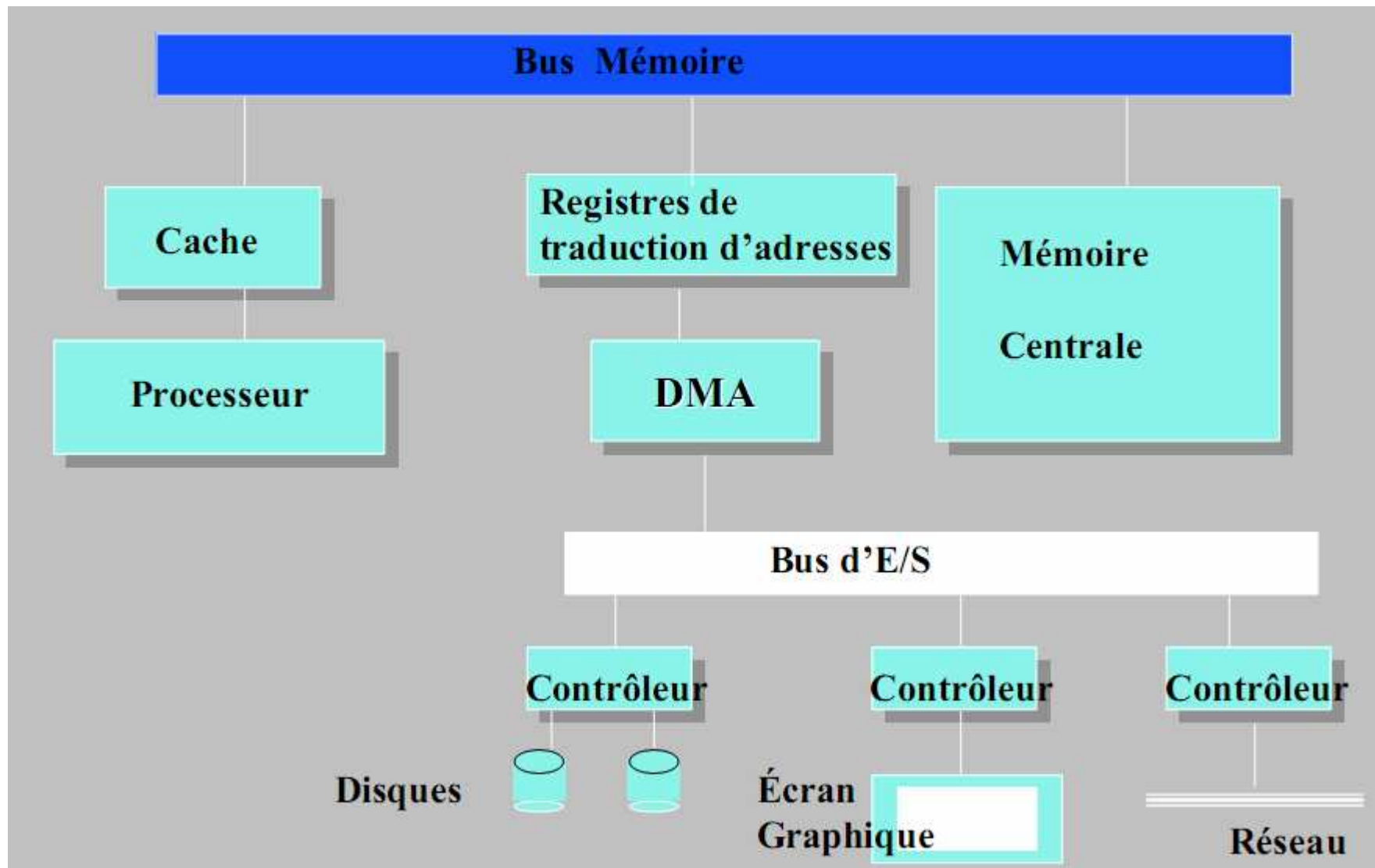
Comment ?

## L'accès direct mémoire (DMA)

- Technique qui fournit un chemin direct entre le bus d'E/S et la mémoire sans intervention du processeur.
- Le DMA soulage l'UC du contrôle des entrées-sorties
- L'E/S "vole" un cycle mémoire à l'UT pour réaliser un transfert
- L'exécution du programme UC continue pendant le transfert DMA
- Si conflit d'accès à la mémoire priorité au DMA



# Accès Direct Mémoire



# Les types d'interruption

## 1- Interne à l'unité de traitement

- \* division par 0
- \* dépassement de capacité numérique
- \* erreur de parité sur la mémoire
- \* erreur de transmission
- \* défaut d'alimentation électrique

## 2- La sollicitation d'une unité périphérique

- \* commande pupitre (RAZ - INIT)
- \* demande de service pour envoyer ou recevoir une donnée

## 3- Les Appels superviseur provoqués par le programme

- \* Lancement de tâches
- \* Allocation de ressources matérielles (mémoire - disque - imprimante ...)
- \* Réalisés par des instructions de type SVC (Supervisor Call)

## 4- Moyen de communication entre unités de traitements dans une architecture multiprocesseur.

# Les Processus

## Critères de découpage en processus concurrents

### ➤ **parallélisme physique présent dans l'application**

- un processus pour chaque organe d'entrée/sortie : imprimante, souris,
- processus clients et processus serveurs dans un système réparti
- processus de lecture de capteurs dans un système embarqué (réseaux de capteurs)

### ➤ **parallélisme logique de l'application**

- activités avec différentes échéances ou importances.
- activités avec différents comportements : périodiques, réactives, cycliques.
- activités de natures différentes : calcul, acquisition, présentation, client, serveur.
- activités distantes, réparties ou mobiles
- regroupement des actions fonctionnellement proches pour limiter les interactions interprocessus.

# Représentation des processus

## ➤ Processus :

- entité d'exécution (séquence d'actions) d'un programme
- exécution sur un processeur
- représentation de ce processus (par une structure de données) dans le moteur d'exécution (le noyau du système)

## ➤ Décomposition d'une application en processus

- statique : le nombre des processus est fixe et connu dès le départ
- dynamique : création et destruction pendant le déroulement

## ➤ Niveau et hiérarchie de processus

- un seul niveau : tous égaux (ex. Concurrent Pascal)
- hiérarchie de processus avec emboîtement (ex. Unix, Ada): relation père-fils et arborescence de la descendance, le père doit attendre que tous ses fils aient terminé.

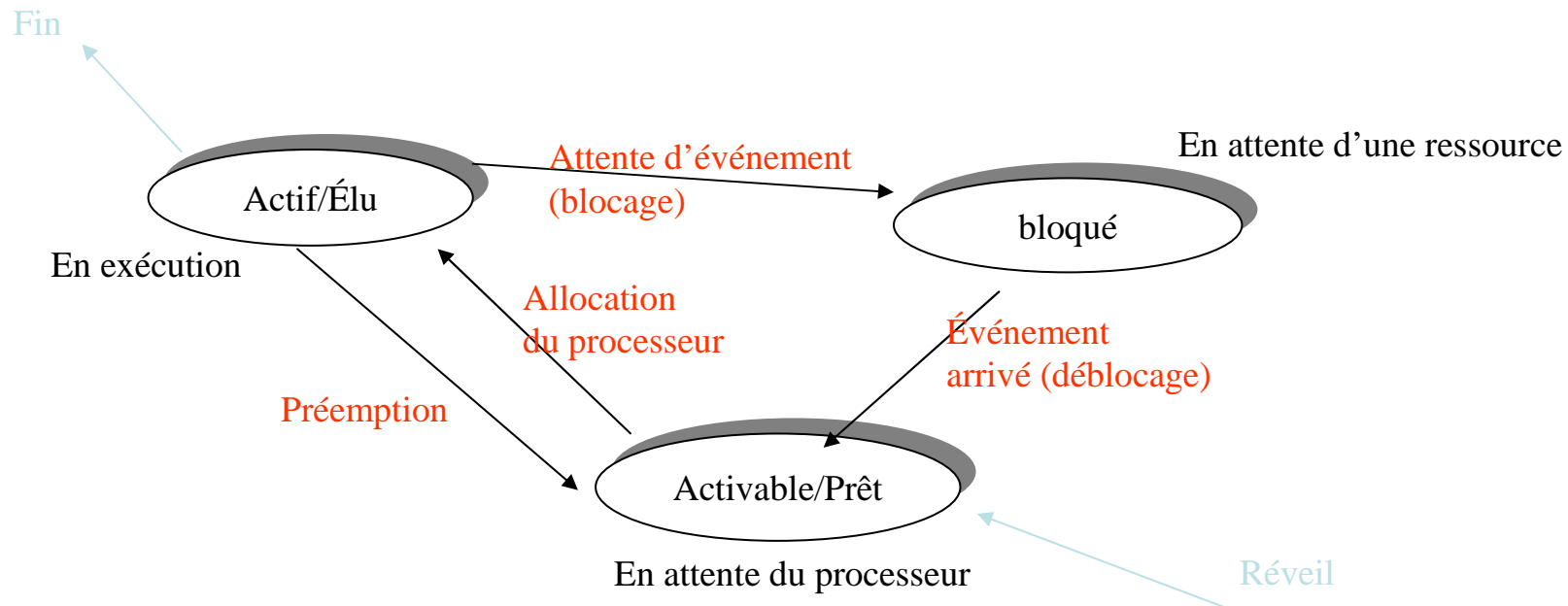
## ➤ Création de processus

- par déclaration d'un modèle (type, classe) et instantiation d'un objet selon ce modèle (ex. Ada, Java)
- par création d'une copie (clone) du père dans son état au moment de la création du fils (ex. Unix, Linux : fork)

# Gestion système des processus

## États d'un processus

- **prêt (activable)**: le processus possède toutes les ressources pour s'exécuter sauf le processeur
- **élu (actif)**: a obtenu le processeur qui exécute les instructions de son code
- **bloqué** : processus bloqué en attente d'un événement (fin d'E/S)



# Création des processus

## ➤ Initialisation de processus

- passage de paramètres à la création (ex. Unix/Linux : **exec** )
- pas de paramètre, donc acquisition explicite par le processus une fois créé (ex. Unix/Linux : **fork** crée un processus avec duplication du code et données)

## ➤ Démarrage ou activation des processus

- à la création (les processus sont créés actifs, en Ada)
- à l'instanciation de classes (création de threads en Java)

## ➤ Mode de comportement des processus: séquentiel, cyclique, périodique, etc.

## ➤ Terminaison de processus

- fin correcte du code
- terminaison imposée par le noyau (erreur, dépassement de ressource)
- auto terminaison (suicide) sur exception
- terminaison forcée par un autre processus (« CTRL/C » sous Unix/Linux)
- jamais : processus cyclique permanent ("démon")
- jamais : boucle infinie par erreur

## Descripteur de processus

**Bloc de contrôle (utilisé par le système) comporte :**

- identificateur repérant de manière unique chaque processus
- état du processus
- environnement volatile (copie des registres du processeur, du compteur ordinal et du pointeur de pile, à la fin de la dernière éléction)
- liens permanents de chaînage vers les descripteurs de ressources



# Ordonnancement des processus

- **Allocation du processeur**
  - sans réquisition possible (préemption) jusqu'à fin de l'état actif
  - avec réquisition si :
    - activation d'un processus plus "prioritaire"
    - baisse dynamique de "priorité"
    - fin de quantum d'allocation
  
- **quelques règles de gestion des processus prêts:**
  - ancienneté (FIFO)
  - quantum de taille fixe ou variable
  - priorités fixes ou dynamiques

# Ordonnancement des processus Linux

Trois politiques d'ordonnancement sous Linux :

- SCHED\_FIFO* : pour ordonnancer des processus non préemptibles
- SCHED\_RR* : pour ordonnancer des processus préemptibles (quantum de temps)
- SCHED\_OTHER* : pour ordonnancer des processus quelconques

-Les processus de la *SCHED\_FIFO* sont plus prioritaires que ceux de la *SCHED\_RR* qui eux sont plus prioritaires que ceux de la *SCHED\_OTHER*.

Les priorités :

- varient de 1 à 99 dans la *SCHED\_FIFO* et la *SCHED\_RR* (le processus de priorité 99 est le plus prioritaire).
- égales à 0 dans la *SCHED\_OTHER*.

# Conclusion

## Le rôle du système d'exploitation

- *Le SE est une interface entre les applications et le matériel*
- Le SE a pour responsabilités de :
  - gérer les ressources (mémoire, disque, périphérique, processeur, etc.)
  - associer les protections nécessaires à la gestion des ressources
  - fournir un accès équitable aux différentes ressources partagées
  - ordonnancer les programmes pour assurer une bonne qualité de service

## Caractéristiques des SE modernes

- Multi-threading
  - Processus légers
- Multi-processus
- Multi-cœur, partage mémoire (multi-processeur)
- Architecture à micro-noyaux (micro-kernel)
  - Gestion de l'adressage
  - Communication interprocess
  - Ordonnancement
- Système distribué

# Bibliographie

