

# MAIN3 : Programmation système

*Principes de base des systèmes d'exploitation*

Cours original de Pierre Sens  
Modifié par **Adrien Cassagne**

[adrien.cassagne@lip6.fr](mailto:adrien.cassagne@lip6.fr)

# Organisation de l'UE

---

- [https://lip6.fr/adrien.cassagne/epu\\_main3\\_sys.html](https://lip6.fr/adrien.cassagne/epu_main3_sys.html)
- 5 séances de 4h (= 20h, nombre d'heures inférieur aux précédentes années)
  - Cours
  - TD
  - TP (un peu de travail à la maison pour terminer les TP)
- Notations :
  - 30 % 1 TP noté (à déterminer)
  - 70 % examen (30/05 de 10h45 à 12h45, amphi)

# Bibliographie

---

- Pré requis
  - Connaitre un peu de C
  - Connaitre un peu l'architecture des ordinateurs
- Livres
  - Principes [appliqués] des systèmes d'exploitation (plusieurs éditions)  
Peter Galvin, Abraham Silberschatz
  - Systèmes d'exploitation (plusieurs éditions)  
Andrew Tanenbaum

# Rôle de l'UE

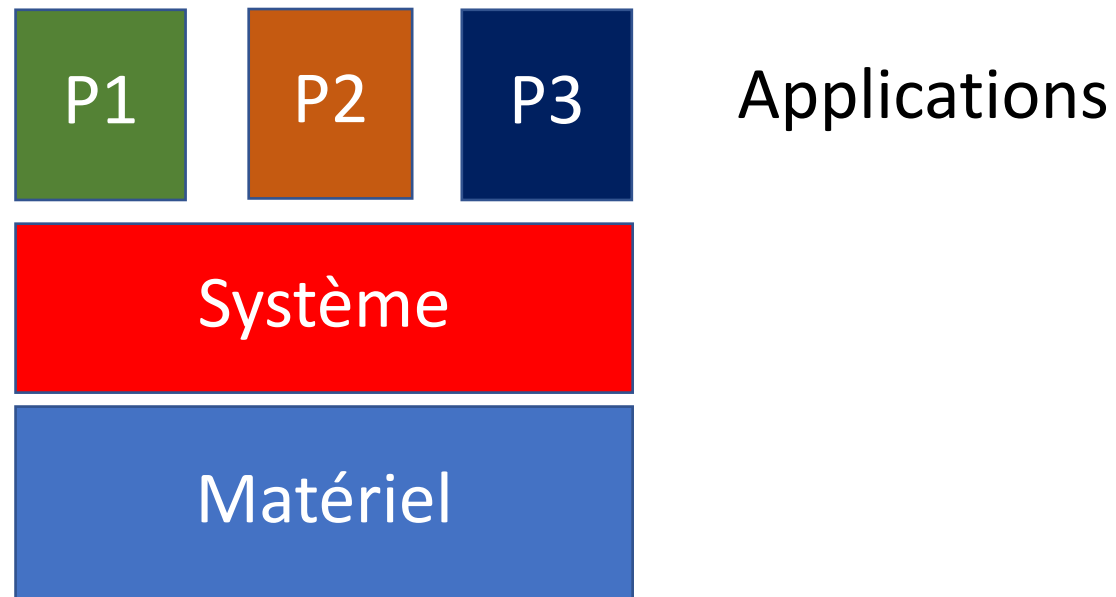
---

- Comprendre les **mécanismes internes et algorithmes** des systèmes d'exploitation
- Pas spécifique à un système (Linux, Windows, BSD ...)

# Cours 1 : Introduction aux systèmes d'exploitation

---

- Définition : un système d'exploitation (Operating system – OS) est un **logiciel** qui gère la **machine physique**
- Offre aux applications une vision abstraite de la machine



# Historique

---

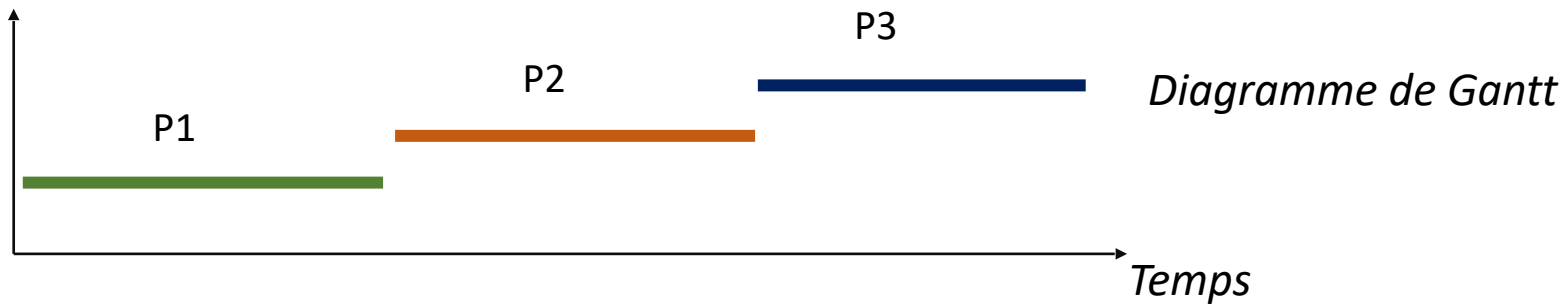
- Les systèmes ont suivi les évolutions des ordinateurs
- **1940** : Ordinateurs à tubes à vide
  - **Pas de systèmes**, Programmation par **recablage**
  - ENIAC : 1946, ~18 000 tubes, 30 tonnes
- **1960** : Ordinateurs à transistors
  - Systèmes centraux = mainframe ("gros" systèmes)
  - Introduction des principaux concepts des OS



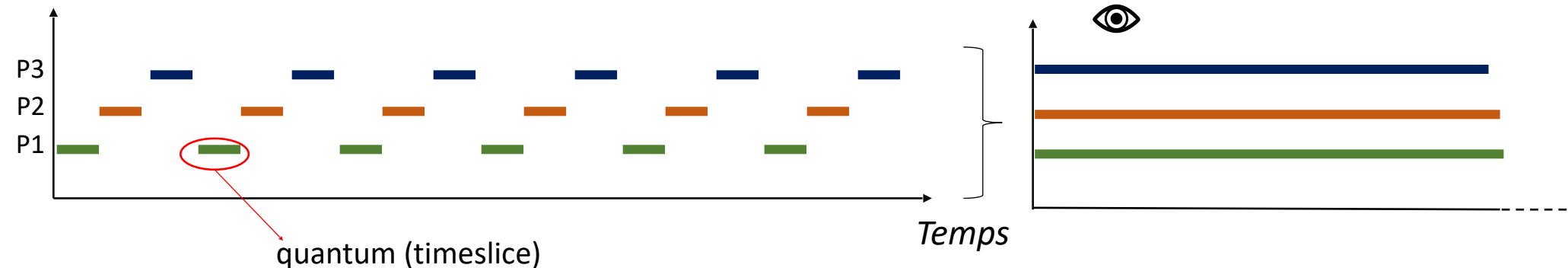
# Historique (2)

- Mainframe

- ATLAS - 1960 : introduction du **traitement par lots (batch)**
- Exécution automatique d'une série de programmes (commandes)

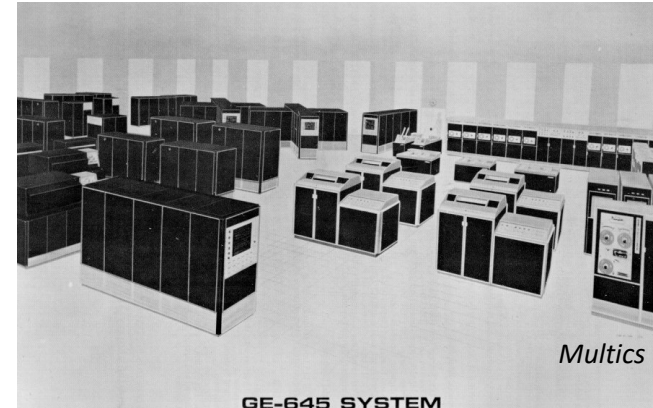


- CTSS (MIT) - 1962 : 1<sup>er</sup> Mainframe à **temps partagé** (time sharing)
- Partage du temps en quantum : temps maximum d'exécution consécutive



# Historique (3)

- Principaux mainframes
  - IBM OS/360 – 1965
  - **Multics** (MIT) - 1964
  - De nombreux utilisateurs et terminaux
- 1965-1970 : Mini-ordinateurs
  - PDP 7/8/11 de DEC
  - Moins puissant, moins d'utilisateurs
  - **Ken Thompson** et Dennis Richie conçoivent un système plus léger écrit en assembleur : UNICS (69) => **UNIX** (71)
- 1973 : Invention du langage C par Dennis Richie
  - UNIX réécrit à 90 % en C



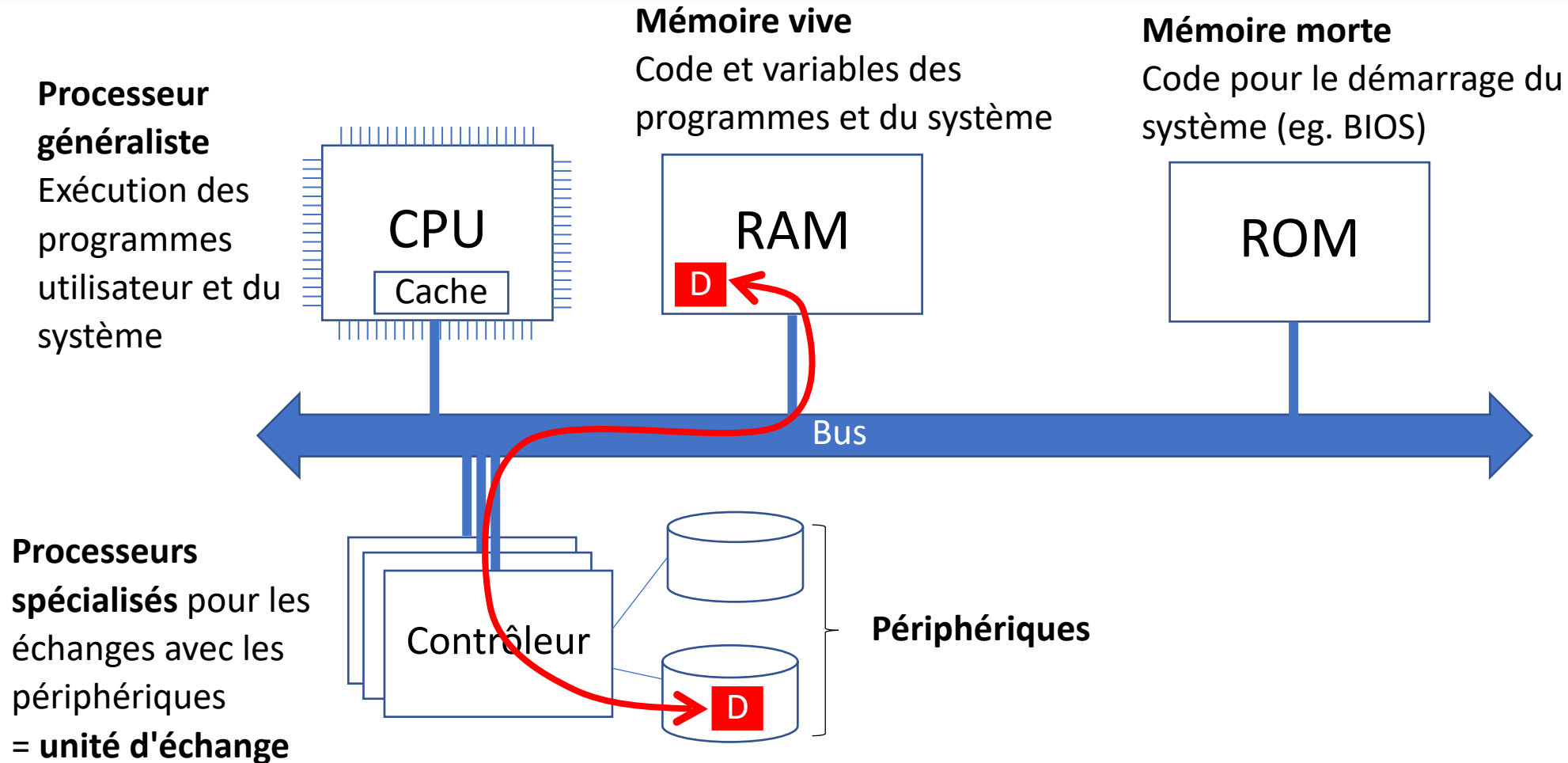


# Historique (4)

---

- **1975-1980** : Micro-ordinateur
  - Machine moins puissante dédiée à **un utilisateur**
  - Apple I - 1976
  - PC (Personal Computer) introduit par IBM - 1981
  - Systèmes : CP/M puis MSDOS (Bill Gates). **Mono-tâches, mono-utilisateurs**
  - Branche MSDOS : ...Windows 3.11 → W.95 → W. 98 → W. Millenium (Fin)
- **1990** : Unix pour PC
  - Minix / Linux / FreeBSD ...
  - Open-source, **multi-tâches, multi-utilisateurs**
- **1993** : Nouvelle branche pour Windows
  - Nouveau système d'exploitation inspiré de VMS (Mainframe)
  - Windows NT : WNT 4 → WNT 5 → W. 2000 → W. XP → W. Vista → W. 7 → W. 8 → W. 10

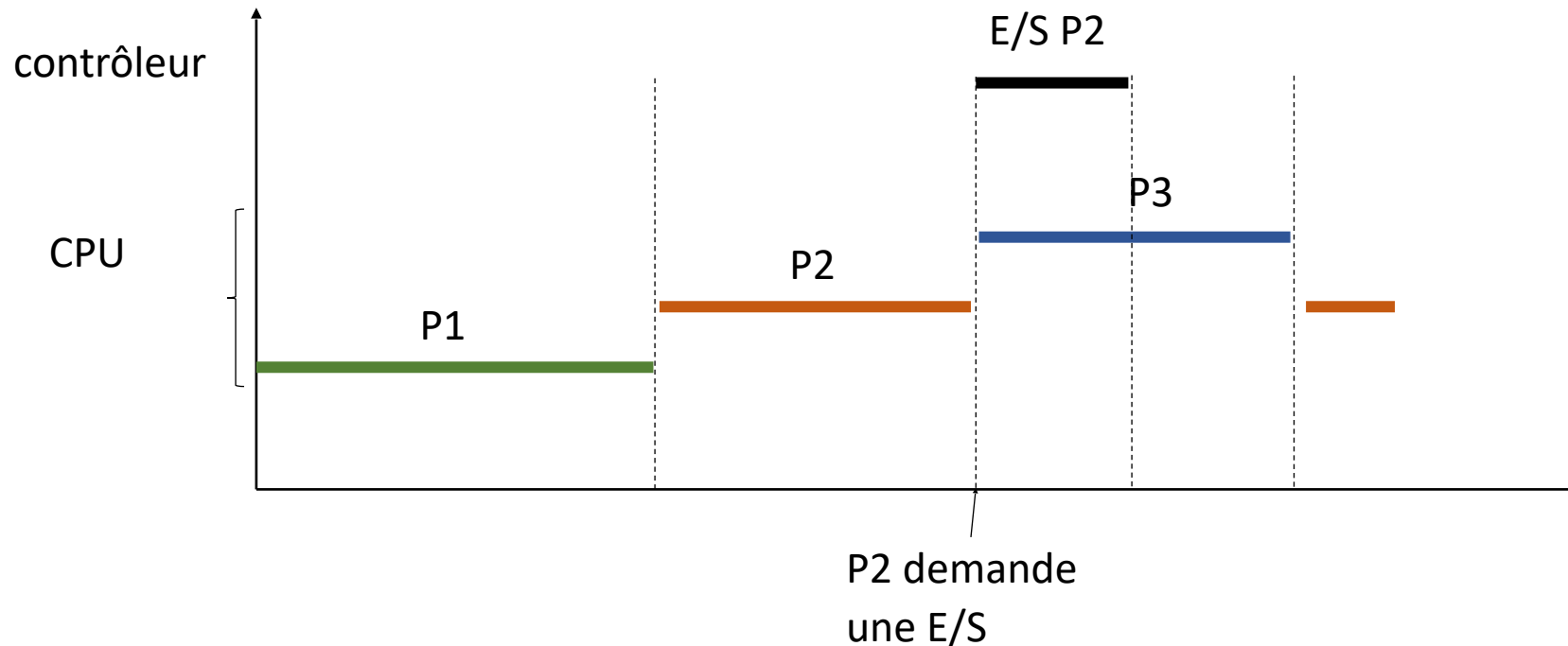
# Composants d'un ordinateur



1 contrôleur par type de périphérique (disque, réseau, ...)  
Accès direct à la mémoire : **DMA** (Direct Memory Access)  
**Entrée/sortie** (E/S) : entrée = lecture, sortie = écriture

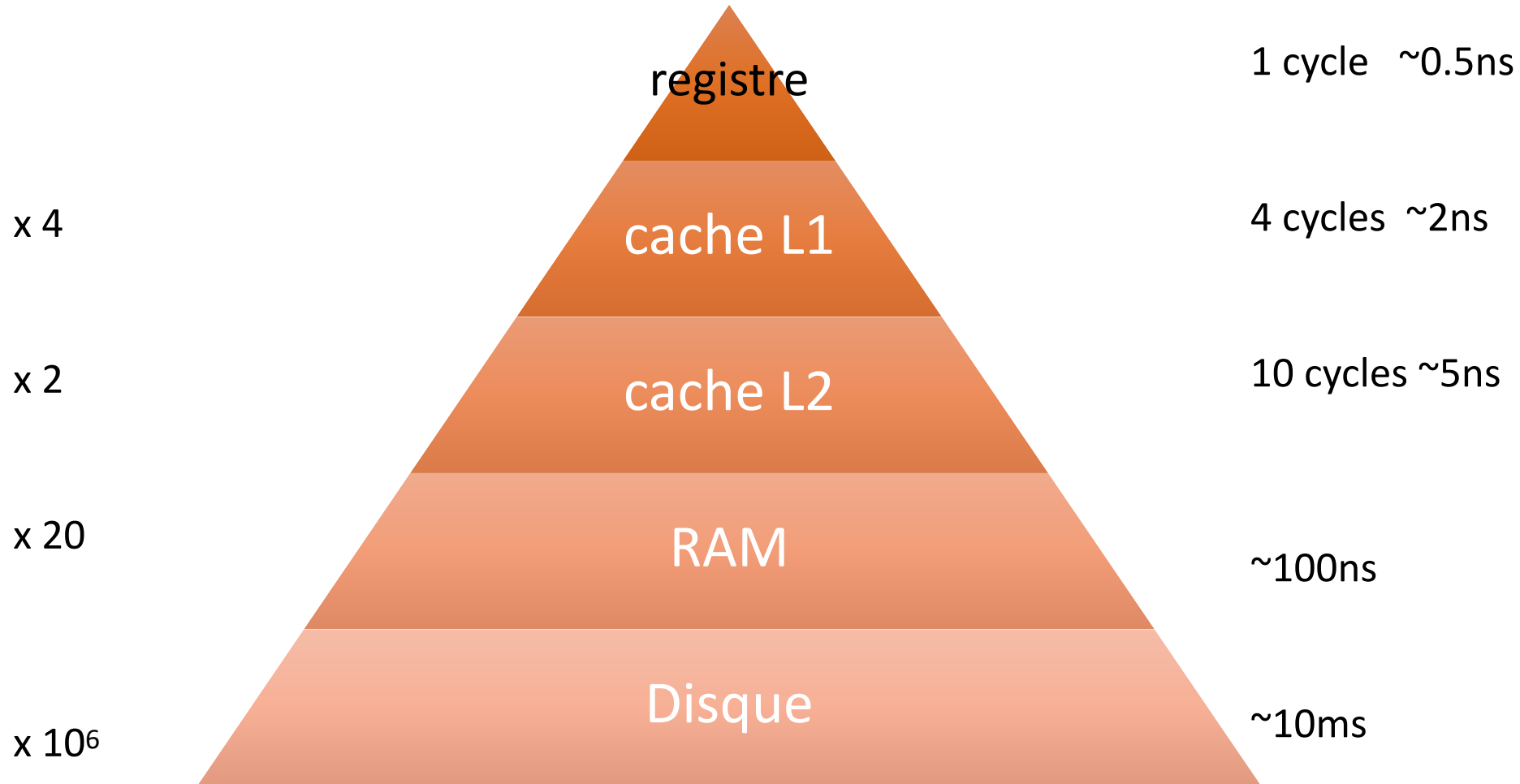
# Composants d'un ordinateur

- Hypothèse du cours : 1 seul cœur par CPU => pas de "vrai" parallélisme entre l'exécution des programmes (tâches)
- Parallélisme entre exécution et transfert depuis les périphériques

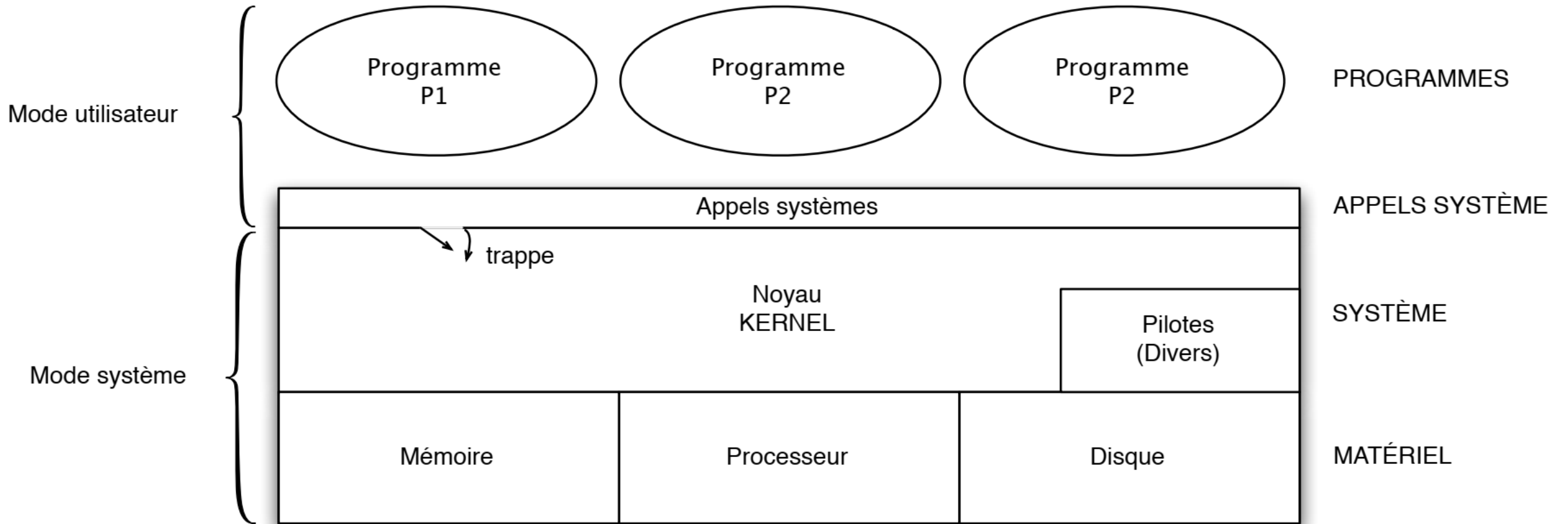


# Hiérarchie des temps d'accès

---



# Systemes d'exploitation : architecture

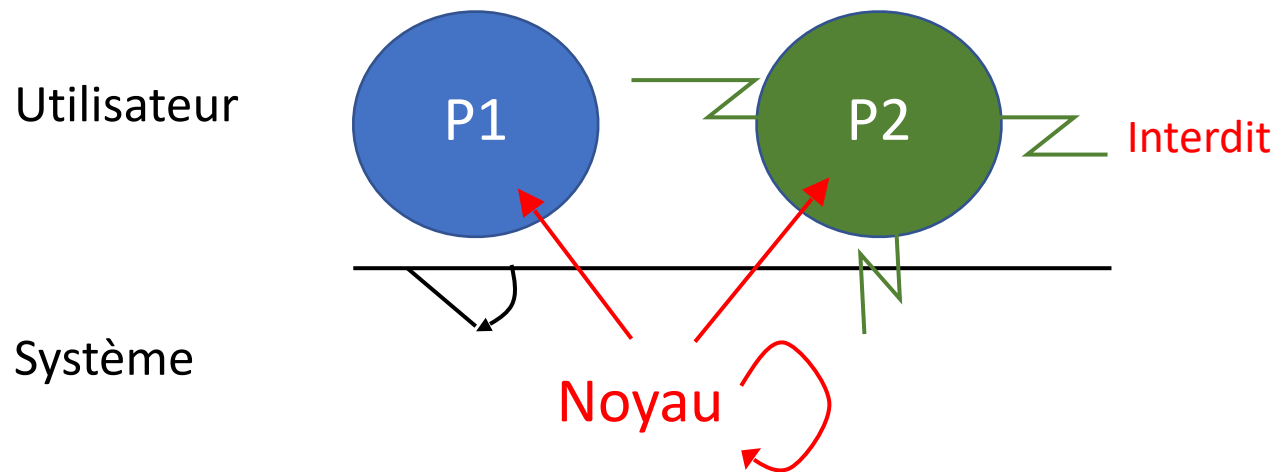


# Modes d'exécution

---

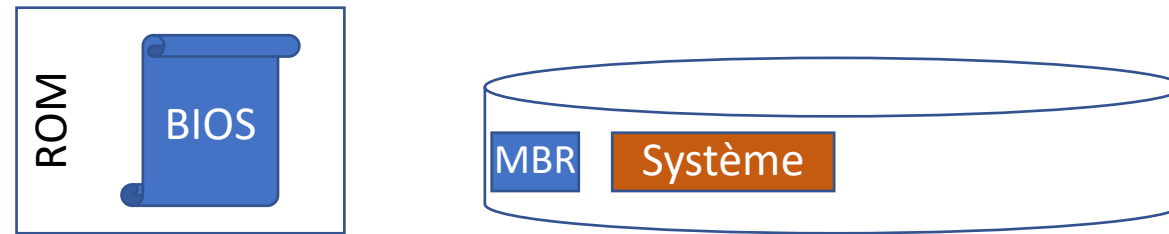
2 modes

- Mode utilisateur : accès restreint (confinement) aux ressources  
Exécution des codes des programmes
- Mode système : accès privilégié à toutes les ressources du système  
Exécution des codes du noyau.



# Amorçage du système (boot)

- A la mise sous tension la RAM est vide : charger et initialiser le système en RAM  
⇒ Utilisation d'un "mini" OS présent en ROM (le BIOS sur PC)



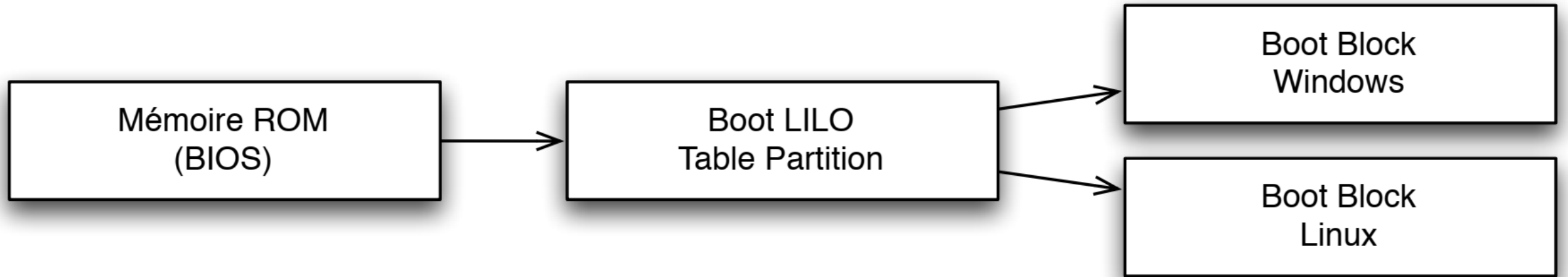
## Séquence d'amorçage

1. Mise sous tension : exécuter le code présent en ROM (BIOS)
  2. Vérifier les périphériques
  3. Copier le contenu du premier bloc du disque en RAM (MBR – Boot loader Block)
  4. Exécuter le boot loader en RAM
  5. Charger le système d'exploitation
- BIOS
- Loader

# Multi-boot

---

- Remplacer le block boot par un code faisant un aiguillage





# Rôles du système

---

Gérer / partager les différentes ressources entre les programmes

Gestion du **processeur** : algorithme d'ordonnancement de tâches

Gestion de la **mémoire** : allocation mémoire, gestion de la mémoire virtuelle

Gestion du **disque** : cache disque et système de gestion de fichiers

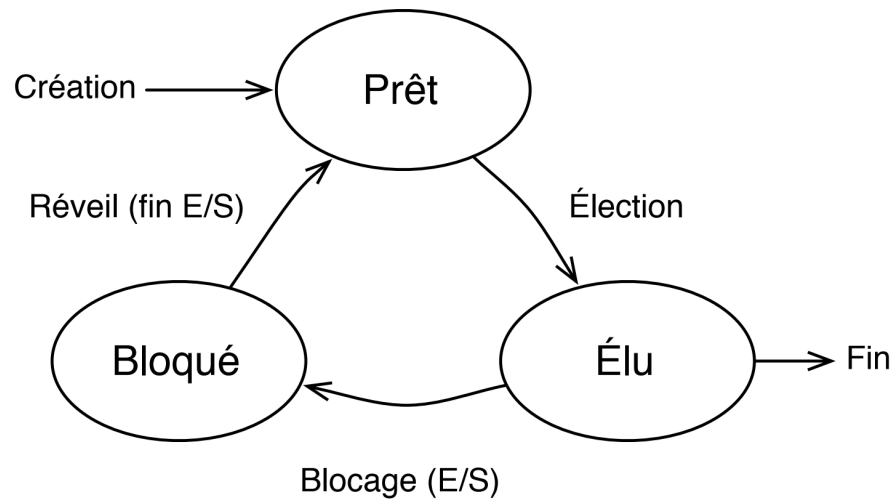
# Définitions

---

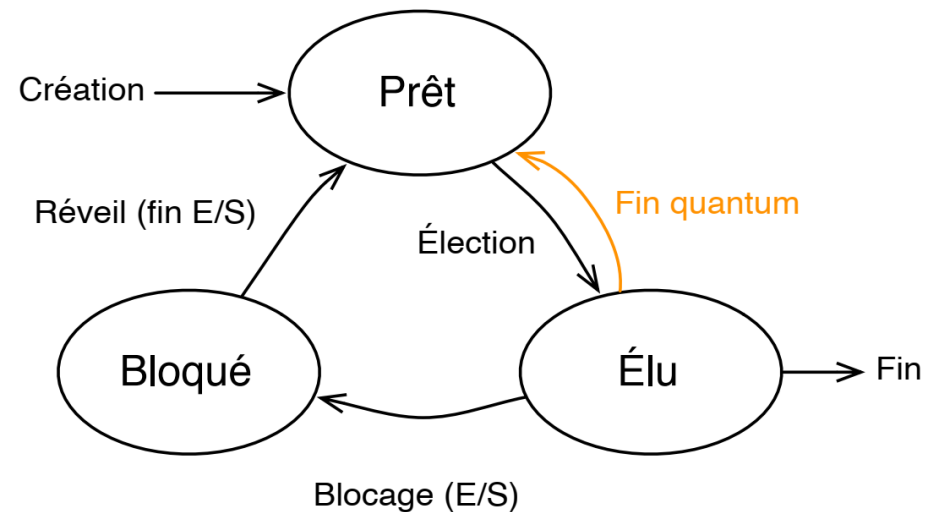
- Programme : exécutable stocké sur disque
- Processus / tâche : programme chargé en mémoire (contexte mémoire et registre) pour être exécuté
- Mono-programmation : 1 seule tâche présente en mémoire
- Multi-programmation : plusieurs tâches présente en mémoire

# Etats d'un processus

- **Prêt** : le processus prêt à s'exécuter
- **Élu** : le processus s'exécute
- **Bloqué** : le processus attend une ressource (E/S, clavier, paquet réseau, un synchronisation ...)



**Batch**



**Temps partagé**