



# Système d'exploitation avancé

Structuration et système d'exploitation

Pierre LEROY – [leroy.pierre1@gmail.com](mailto:leroy.pierre1@gmail.com)

# Sommaire

---

- I. Architecture
  - II. Processeur
  - III. Typologie SE
  - IV. Conclusion
-

# Architecture

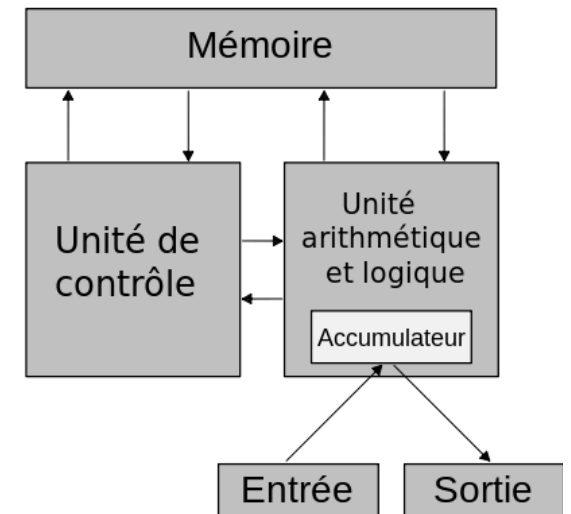
- Un ordinateur s'articule autour des principes décrits par Von Neumann :

## ARCHITECTURE « VON NEUMANN » (1945)



### COMPOSANTS

- **UAL** (*ALU*) – Unité Arithmétique et Logique
  - ou unité de traitement
  - effectue les opérations arithmétiques de base.
- **UC** – Unité de Contrôle
  - en charge du séquençement des opérations
- **MEM** – Mémoire
  - contient à la fois les données et le programme
    - Distinction entre mémoire vive et mémoire de masse
- **E/S** – Entrées-Sortie
  - Dispositif de communication avec l'extérieur

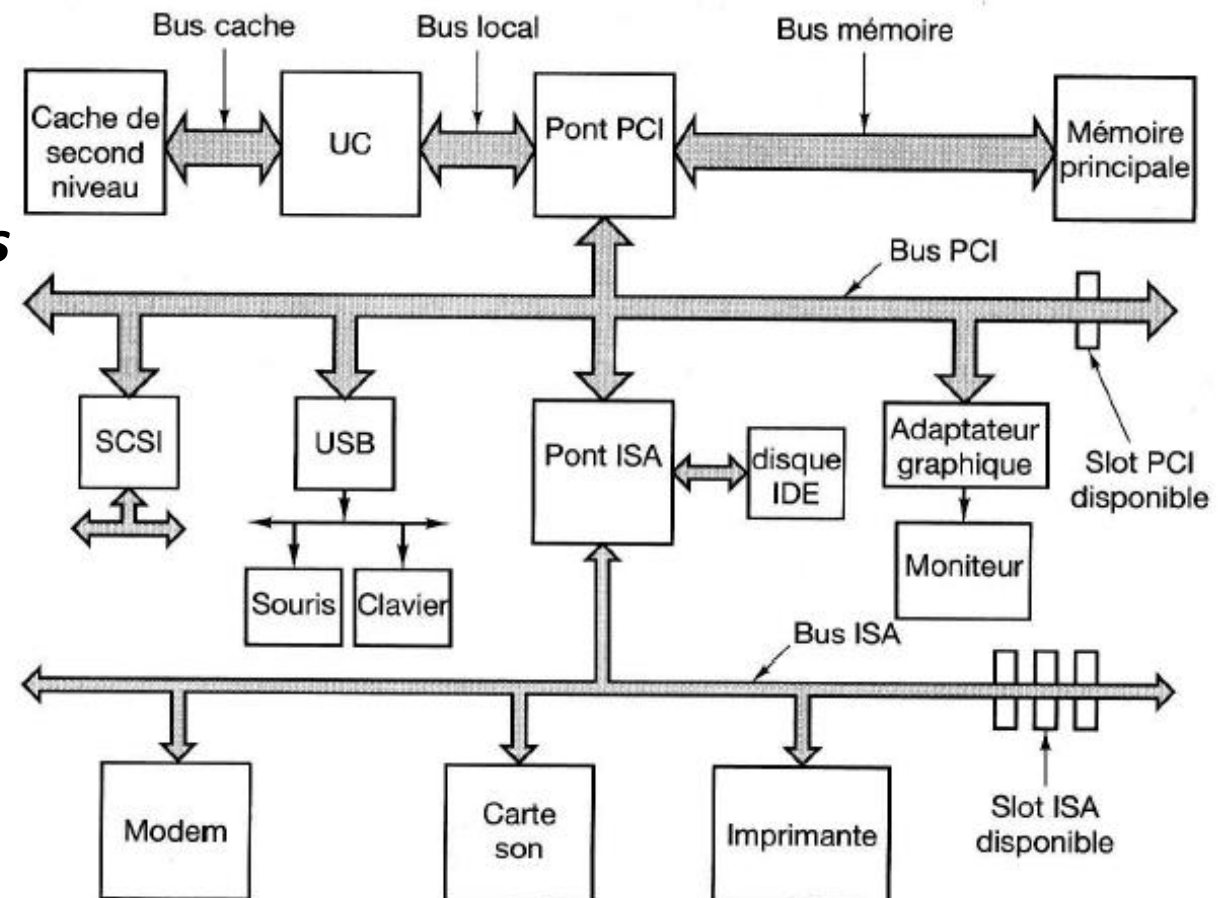


# Architecture

- Un ordinateur s'articule autour des principes décrits par Von Neumann :

## ARCHITECTURE « Pentium » :

- Logique semblable
  - Multiplication des **bus**
  - Multiplication des **contrôleurs**



# Sommaire

---

- I. Architecture
  - II. Processeur
  - III. Typologie SE
  - IV. Conclusion
-

# Processeur

## GENERALITES

- **Extrait des instructions** de la mémoire et les **exécute**.
- Possède un **jeu d'instructions** qui lui est **propre**.
- **Registres** : stockent des variables importantes et des résultats intermédiaires.
- **Instructions** classiques :
  - charger un mot de la mémoire vers un registre (et inversement).
  - opération arithmétique : une ou deux opérandes, stockage du résultat dans un registre.
- **Compteur ordinal** : adresse de la prochaine instruction à exécuter.
  - Mis à jour à chaque exécution d'instruction.
- **Pointeur de pile** : adresse courante du sommet de la pile.
- **Mot d'état (PSW, Program Status Word)** : entre autre, mode (utilisateur ou noyau) d'exécution. Rôle très important dans les appels système.
- **Pipeline** : unités séparées pour *l'extraction, le décodage* et *l'exécution*.
  - L'instruction **n+1** est décodée / l'instruction **n+2** est extraite / l'instruction **n** exécutée.
  - Une instruction qui rentre dans le pipeline est exécutée, même si l'instruction précédente était un branchement conditionnel.

# Sommaire

---

I. Architecture

II. Processeur

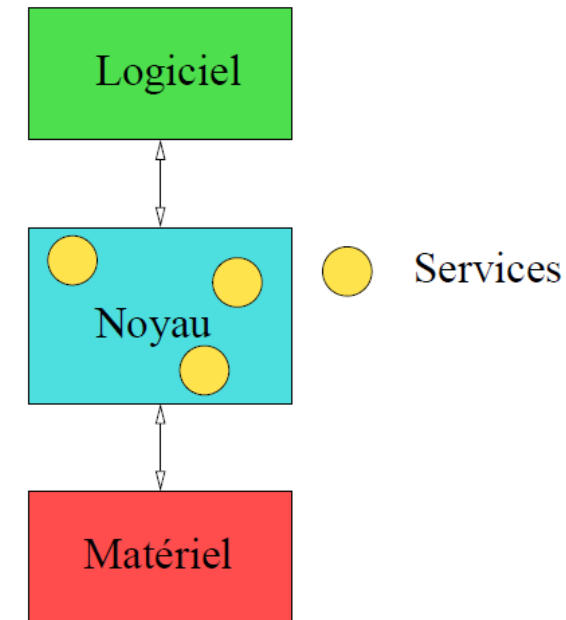
III. Typologie SE

IV. Conclusion

---

# Mode utilisateur/noyau

- Information stockée dans le **PSW**.
- Mode **noyau** : le processeur peut exécuter tout son jeu d'instruction et a accès à l'ensemble du matériel.
- Mode **utilisateur** : sous-ensemble d'instructions.
  - E/S (entrées/sorties) et accès à la mémoire inaccessibles.
  - Interdiction de basculer le bit de protection du PSW.
- Pour accéder aux services du SE, un programme utilisateur doit effectuer un appel système.
- Le programme bascule en mode noyau, puis reprend la main à la fin du traitement de l'appel système.





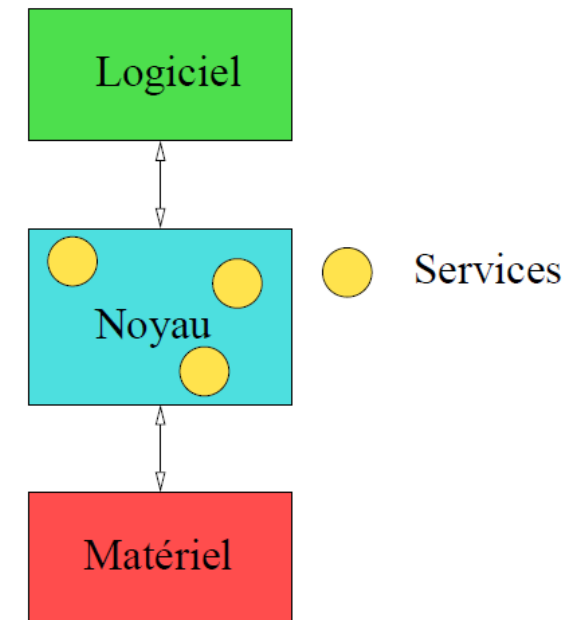
# Noyau & Pilotes

NOYAU

- Partie qui réside constamment en mémoire centrale.
- Routines qui requièrent un mode d'exécution privilégié :
  - mode **superviseur** ou mode **noyau** : appels systèmes.
- Passage en mode noyau nécessite un changement de contexte :
  - Appelé « **commutation de contexte** » : sauvegarde des registres et restauration du contexte.
  - Mécanisme coûteux.

PILOTES

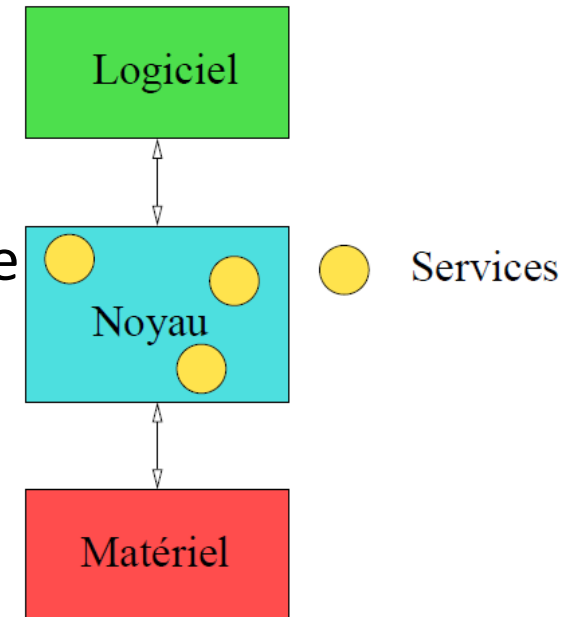
- Modules qui dialoguent avec les périphériques réels.



# Typologie de Noyau : Monolitique

## NOYAU MONOLITHIQUE

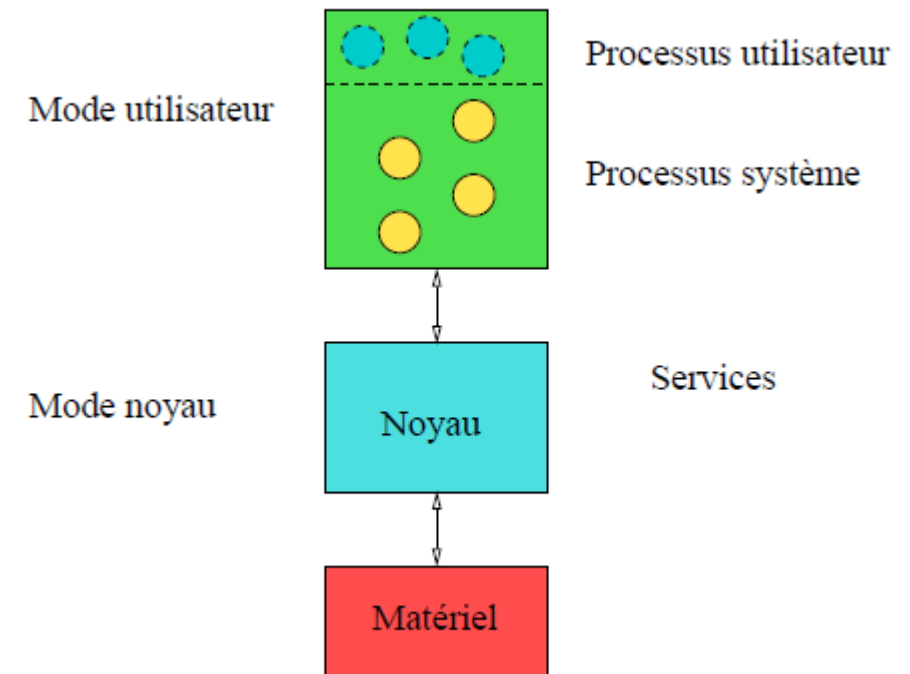
- Anciennes versions d'UNIX, de Linux (<1.2) :
  - Nécessite une recompilation du noyau à chaque mise à jour /installation de pilote.
- Versions récentes de Linux :
  - Tant vers un système modulaire mais reste toujours monolithique
  - Les pilotes sont chargés dynamiquement dans le noyau.
  - Meilleures performances, mais problème de robustesse :
    - ✓ Injecter un pilote buggé dans le noyau peut mettre à mal tout le système d'exploitation.



# Typologie de Noyau : Micronoyaux

## MICRONOYAU

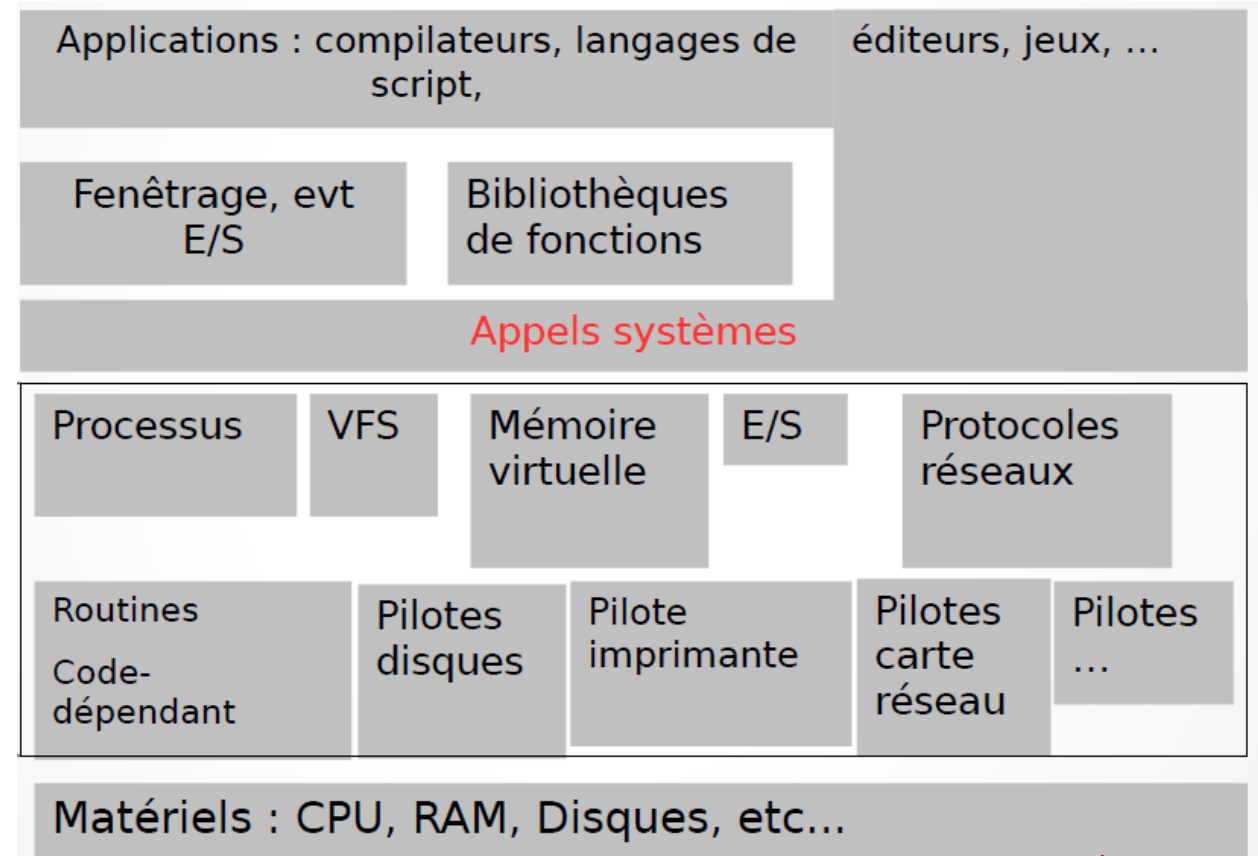
- Noyau minimaliste
- les pilotes sont chargés dans l'espace utilisateur.
- Meilleure extensivité et adaptabilité.
  - Un pilote buggé ne fait pas s'écrouler le système.
- Problème de performances.
  - Systèmes très complexes.
- Exemples:
  - Système GNU Hurd (en développement).
  - Windows NT, MacOS X : pas véritable
  - micronoyau, architecture hybride.



# Constat

- Un système d'exploitation a pour objectif de gérer 5 fonctions essentielles :

- Les processus
- La mémoire
- Les E/S (réseau compris)
- Les fichiers
- Les utilisateurs



# Sommaire

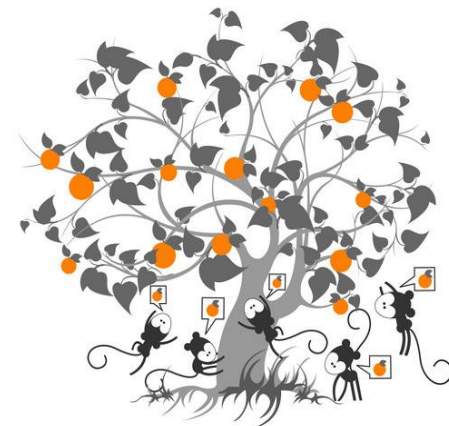
---

- I. Interruptions
  - II. Appels système
  - III. Essentiel
  - IV. Conclusion
-

# Essentiel

---

- Toutes les notions abordées dans ce chapitre sont fondamentales



# Conclusion



# Annexes



# Annexes

---

- Liens annexes :
  - Von Neumann : [https://fr.wikipedia.org/wiki/John\\_von\\_Neumann](https://fr.wikipedia.org/wiki/John_von_Neumann)