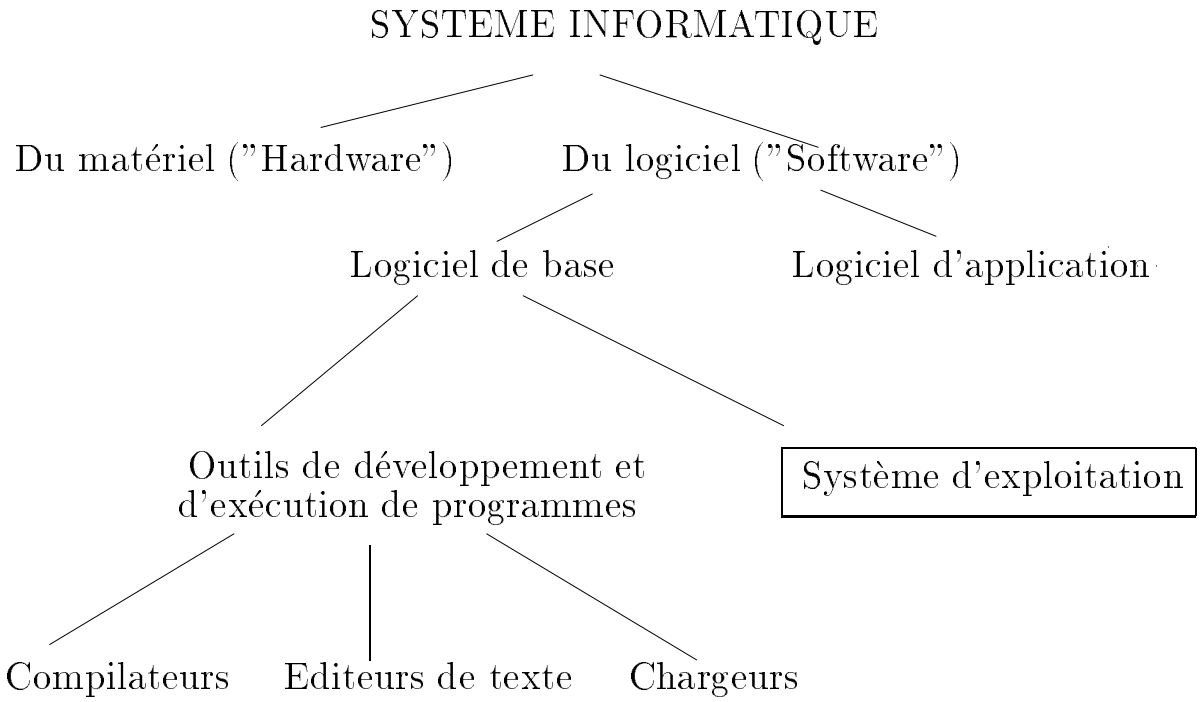
Cours de Systèmes d'Exploitation
Licence d'Informatique
ISTV 98/99

H.Bourzoufi

INTRODUCTION



Qu'est ce qu'un système d'exploitation ? Un SE = Un allocateur et gestionnaire des ressources.

- Les ressources matérielles d'un ordinateur sont :
  - → Unités centrales (UC),
  - → mémoires,
  - → périphériques des Entrées/Sorties
- Pour exécuter un programme il faut des ressources.
- Les systèmes informatiques actuels sont multi-usagers :

  ils exécutent simultanément plusieurs programmes pour le compte de plusieurs programmeurs.

- Les ressources étant limitées :

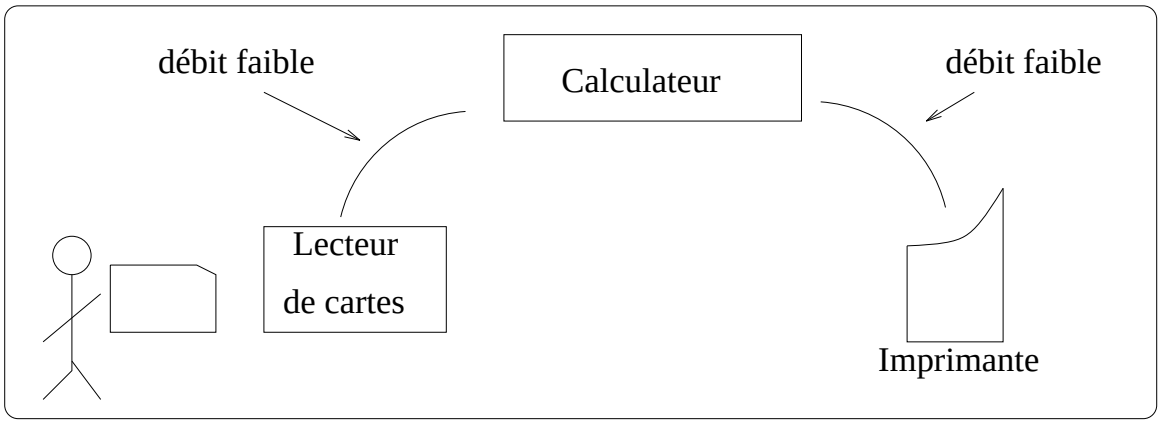
  nécessité de gérer les ressources aux programmes

- Pourquoi les ressources sont elles limitées ?
  - → Motivations de nature économique.
  - → Contraintes de cohérences : partage des informations

Petite histoire des Systèmes d'exploitation

Les premiers systèmes informatiques (1945-1955)

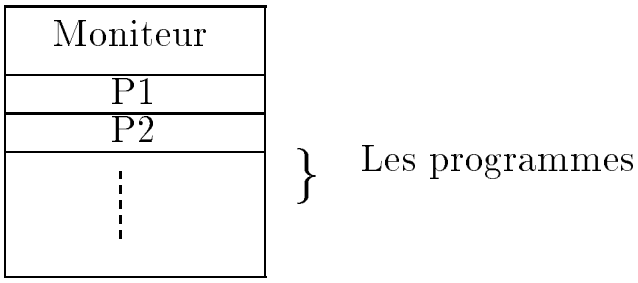
- **Caractéristiques**
  - → du matériel uniquement
  - → pas de système d'exploition
  - → système mono-usager



- **Problèmes**
  - → Gestion du système basée sur la réservation de plages horaires
  - → Manque de fiabilité du matériel
- **Evolution**
  - → Periphériques : apparition des dérouleurs de bandes magnétiques
  - → Logiciel : Apparitions des premiers outils du logiciel de base : assembleur, chargeurs, compilateurs fortran et cobol

Les systèmes à moniteurs(55-65)

- Solution aux problèmes de réservation et de temps de préparation
- **La technique** : Enchainement automatique des programmes par exécution d'un moniteur.



Mémoire de l'ordinateur

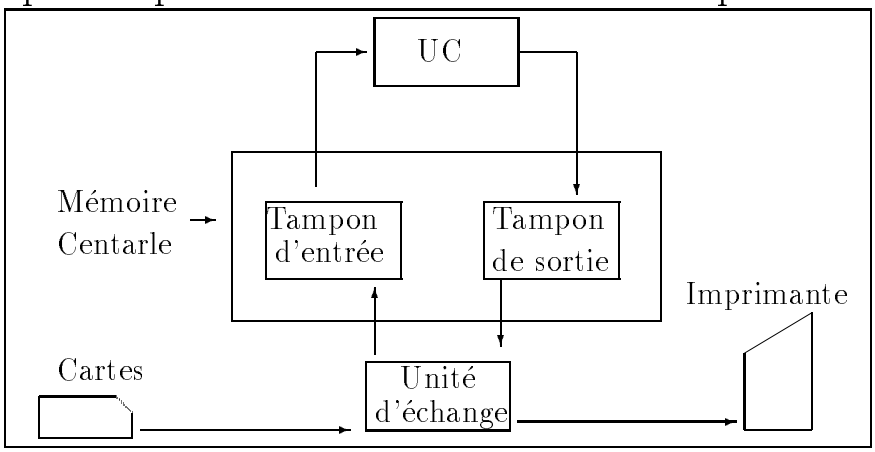
- **Caractéristiques du système**
  - → Système d'exploitation = moniteur
  - → système non interactif
  - → traitement par lot
  - → système multi-usagers
  - → fonctionnement en mono-programmation : Exécution d'1 seul programme à la fois
- **Problèmes de protection**
  - → Comment éviter qu'un programme d'application puisse écrire dans la zone réservée au moniteur ?
  - → Comment forcer le programmeur à utiliser les pilotes de périphériques présents dans le moniteur et lui interdire d'agir directement sur les périphériques?
  - → Comment éviter qu'un travail monopolise l'UC?

Améliorations diverses des systèmes informatiques

Problème: La lenteur des périphériques par rapport à l'UC

- **Les Entrées/Sorties Tamponnées**: Utilisation d'Unités d'échange (UE) capables de fonctionner simultanément avec l'UC.

  → **Principe**: Les cartes sont lues par l'UE et stockées dans des tampons(buffers) d'entrée. L'UC lit les données dans le tampon et produit le résultat dans le tampon de sortie.



  → **Problèmes**:

  ⋆ Ajout et retrait simultanés dans le tampon

  ⋆ Encombrement de la mémoire

- **Les Entrées/Sorties spoolées**

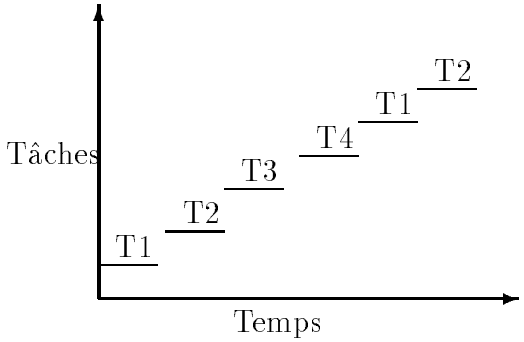
  → **Principe**: Tampons en mémoire secondaire

  → **Problème**: l'UC est contrainte à attendre la terminaison des opérations d'E/S

  → **Solution**: La multiprogrammation :Quand l'UC se trouve en attente d'E/S, elle suspend le programme en cours et reprend l'exécution d'un autre programme :Donc plusieurs programmes résident simultanément en mémoire

Les systèmes à temps partagé

- **Principe :** Considérer que l'UC est une ressource et l'allouer durant un temps limité = partage de l'UC



=> Systèmes interactifs multi-usagers fonctionnant en multiprogrammation avec partage de l'UC

=

Systèmes à temps partagé "time sharing"

Exemple : UNIX

- **Problèmes :**

→ Gestion des périphériques

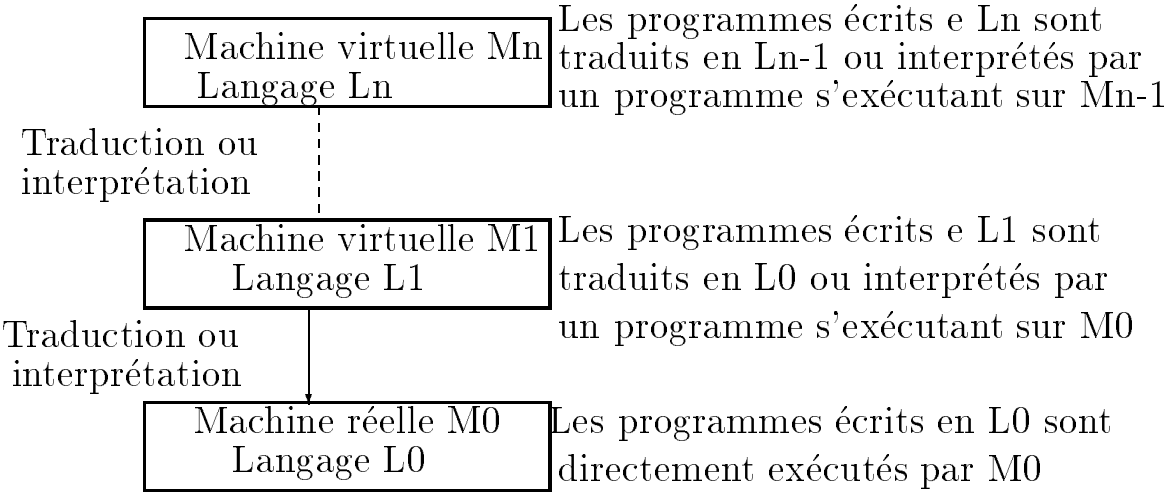
→ Gestion des mémoires (centrale et secondaires)

→ Gestion des erreurs

Nécessité d'un ensemble de programmes (Système d'exploitation) pour résoudre ces problèmes.

Un SE= Une machine virtuelle

- **SE = Réalisation d'une machine virtuelle** au-dessus de la machine matérielle permettant au programmeur de s'abstraire des détails de mise en œuvre du matériel.
- **Notion de machine virtuelle**

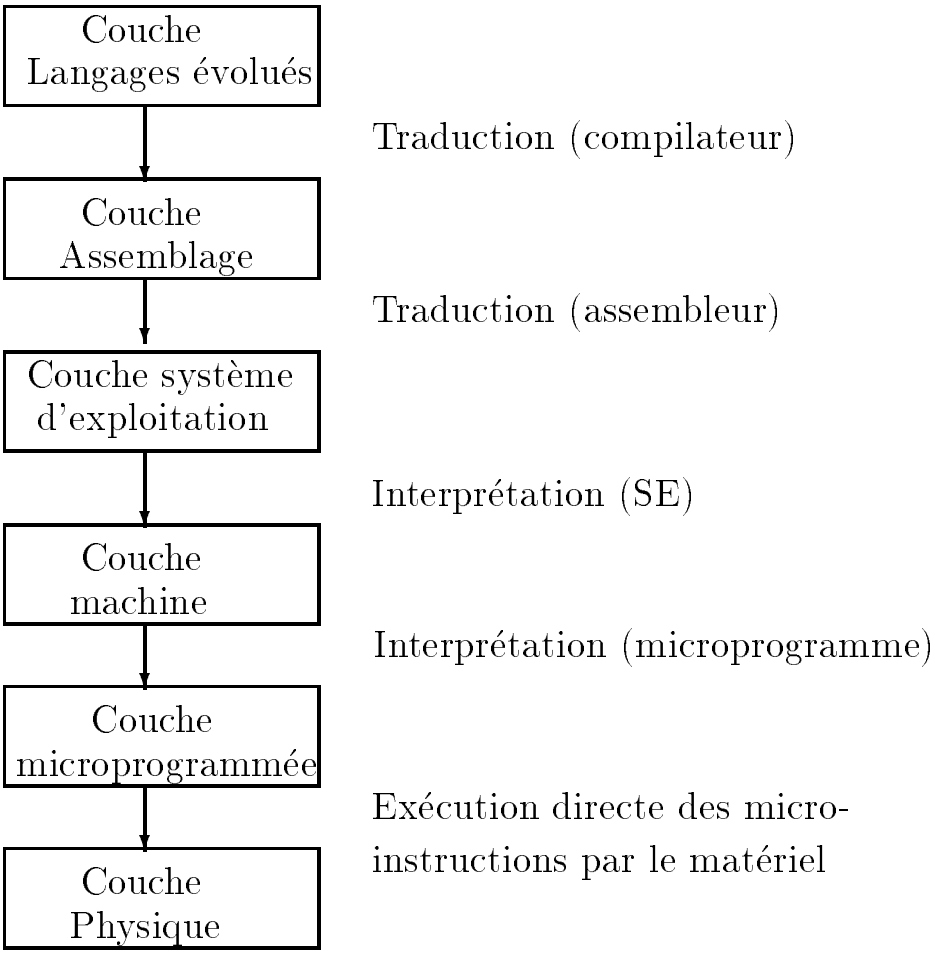


→ **Traduction** : Analyser chaque instruction d'un programme P_i écrit en L_i et la remplacer par la séquence d'instructions équivalentes dans le langage L_{i-1}.

→ **Interprétation** Ecrire, dans le langage L_i, un programme I capable d'analyser, une à une, chaque instruction d'un programme écrit en L_{i+1}, et exécuter immédiatement la séquence d'instructions L_i équivalentes. I est appelé interpréteur.

→ **Seule contrainte** : "respect de la hiérarchie " Un programme s'exécutant sur la machine M_i ne peut être traduit ou interprété en instructions d'un langage L, tel que L soit associé à une machine M_j avec $i < j$

Les systèmes multicouches

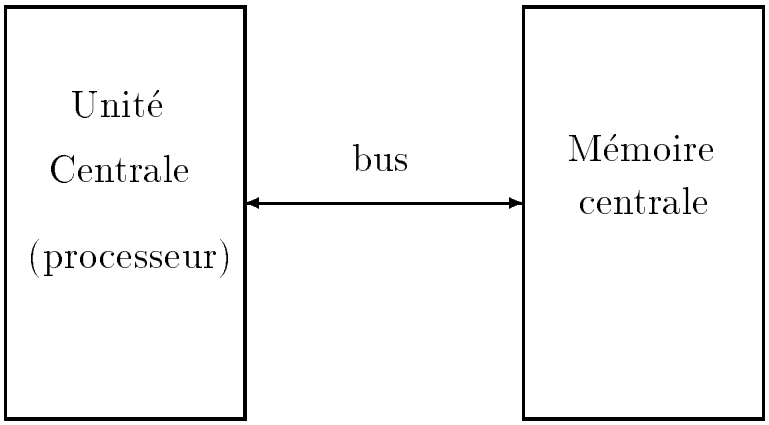


En résumé : On peut dire qu'un ordinateur peut être vu comme un ensemble de couches, chaque couche englobant toutes les couches de niveau inférieur. Une couche représente un certain niveau d'abstraction et comporte divers objets et opérations sur ces objets.

L'ensemble des types de données, des opérations et des caractéristiques de chaque niveau s'appelle l'architecture de ce niveau.

Architecture de la couche physique

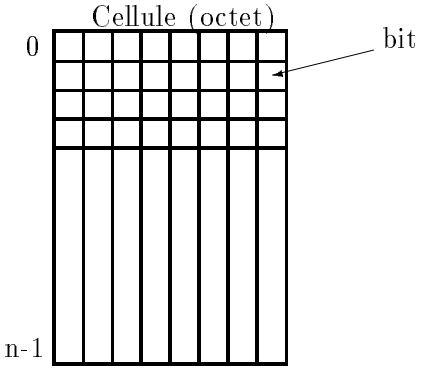
- Modèle classique (modèle von Neumann 1945)



- Le rôle de l'unité centrale (UC) est d'exécuter les programmes stockés dans la mémoire principale.
- La mémoire contient les programmes et données :
  → les mémoires volatiles (RAM : Random Access Memory)
  → Les mémoires mortes (ROM : Read Only mémory)

La Mémoire centrale

- La mémoire est la partie de l'ordinateur dans laquelle programmes et données sont rangés
- Les informations (codes) sont stockés sous forme binaire
- La mémoire peut être vue comme une grille ou chaque case mémorise un chiffre binaire (0 ou 1) appelé **bit**



- Un mot est constitué de 1 ou plusieurs octets consécutifs
- Chaque mot à un numéro appelé **adresse**
- Capacité de la mémoire = nombre total de bits

  → Si les adresses sont sur k bits, on peut adresser 2^k mots

  → la capacité de la mémoire est $8x2^k$ bits = 2^k octets

  → Exemples :

  ⋆ k=8, capacité = $8x2^8$=2048=256 octets

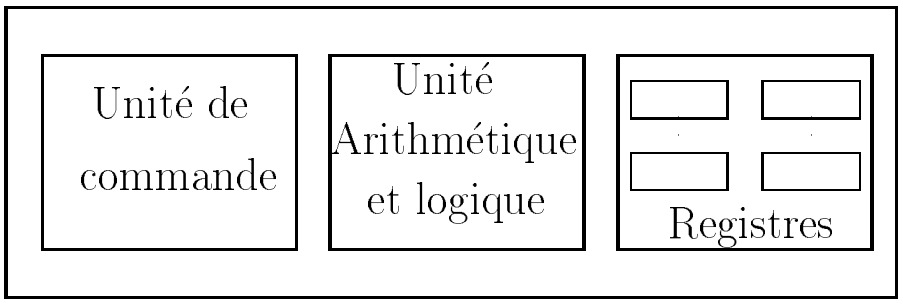
  ⋆ k=16, capacité = 2^{16} octets = 64 x 2^{10}=64 K

  ⋆ 1K (Kilo octets)=2^{10}=1024 octets

  ⋆ 1M (Méga octets) = 2^{20} , 1G (Giga octets)= 2^{30}

Structure interne d'un processeur

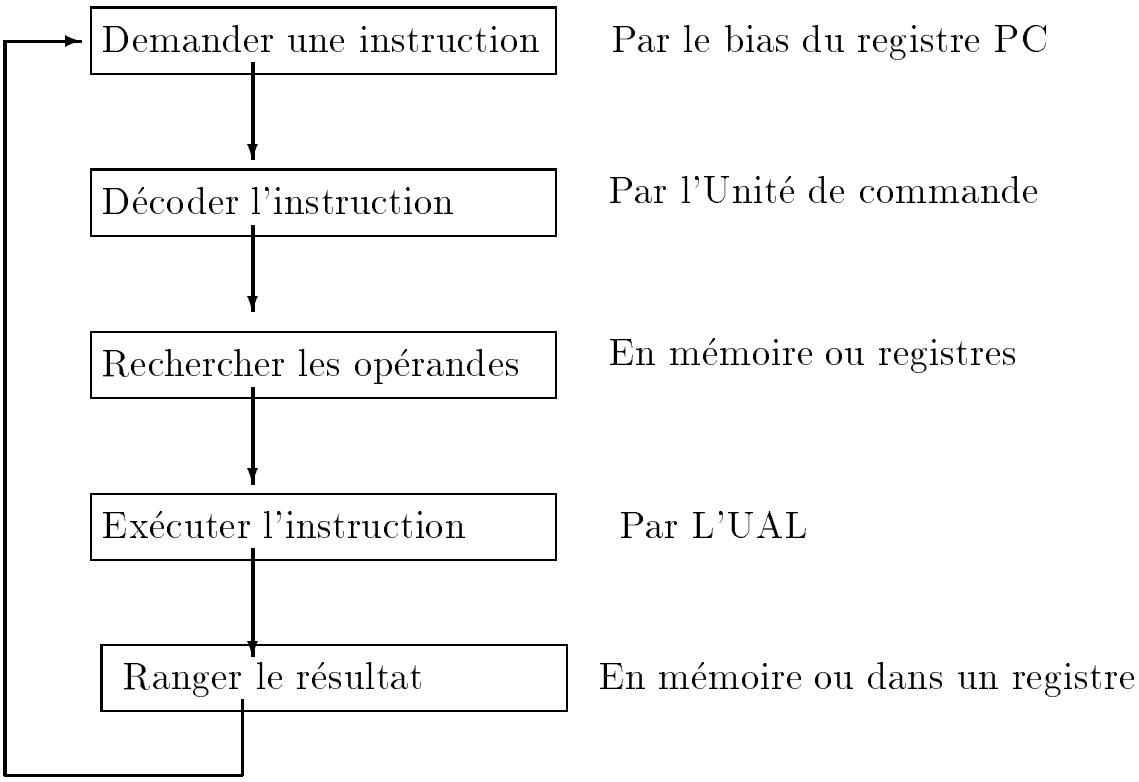
- **Exemples de processeurs :**
  - → Famille Intel : 80xx 80xxx pentiums
  - → Famille Motorola : 68xxx sparcs
- **Jeu d'instructions d'un processeur** = langage machine : C'est l'ensemble d'instructions que peut exécuter le processeur C'est le langage de programmation de plus bas niveau
- **Les instructions sont de type :**
  - → additions de 2 nombres
  - → tests (très élementaires)
  - → mémoire (écrire et lire un nombre en mémoire)
- **Structure interne d'un processeur :**



  - → **Unité de commande :** Charge une instruction et la décode
  - → **UAL :** Exécute les opérations
  - → **Les registres :** Mémoires à accès très rapide qui permettent de stocker des résultats temporaires ou des informations de contrôle

Exécution d'un programme

- **Pour exécuter un programme, l'UC dispose :**
  - → d'un registre **PC** (Compteur ordinal ou de programme), il indique l'endroit en mémoire principale de la prochaine instruction à exécuter.
  - → d'un registre d'instruction **RI** qui contient le code del'instruction à exécuter.
  - → d'une Unité Arithmétique et Logique (ALU ou UAL)
  - → de diverses registres
- **Exécution d'un programme :**



Les systèmes actuels

- **Les systèmes des ordinateurs personnels (PC)**
  → Mono-usager et fonctionnent en mono-programmation
- **Les systèmes à temps partagé**
- **Les systèmes de commandes de procédés**
  → Phériphériques + Capteurs ...
  → Contrainte de temps réel : Le temps de réponse est borné (très court) garanti quelque soit l'activité du système
- **Les systèmes à transaction**
  → Gèrent des bases de données de grande taille
  → Mise à jour de la base par des transactions
- **Les systèmes multiprocesseurs**
  → **But :** Hautes performances
  → Le système gère l'allocation de plusieurs UCs
- **Les systèmes répartis :** facilite l'exécution répartie
  Le but des systèmes répartis est :
  → Partage des ressources , Accélération du calcul
  → Fiabilité et Communication
- **Les systèmes réseaux** Un système réseau permet aux utilisateurs des stations de travail reliés par un réseau de partager des ressources communes, par exemple un système de fichiers.
  Exemple de système réseau : NFS (Network File System)