

---

## 1 & 2. Structure interne et Composants d'un ordinateur

Un ordinateur est une machine qui résout des problèmes en exécutant un programme (une suite d'instructions). Avant exécution, ce programme doit être converti en code machine.

L'ordinateur est composé de trois éléments principaux :

+2

- **Le Processeur (CPU)** : C'est le cerveau. Il charge, décode et exécute les instructions. Il contient l'Unité de Commande (génère les signaux de commande), l'UAL (Unité Arithmétique et Logique pour les calculs) et des Registres (mémoires très rapides pour stocker les résultats temporaires) .

+1

- **La Mémoire principale** : Stocke les programmes et les données.

+1

- **Les Dispositifs d'Entrée/Sortie (E/S)** : Permettent la communication avec l'extérieur.

+2

## 3. Étapes d'exécution d'une instruction

L'exécution suit un cycle "Chargement-Décodage" (Fetch-Decode) précis:

1. Chargement (Fetch) de l'instruction depuis la mémoire vers le Registre d'Instruction (RI).
2. Modification du Compteur Ordinal (CO) pour pointer vers l'instruction suivante.
3. Décodage de l'instruction.
4. Localisation et chargement des données nécessaires dans les registres.
5. Exécution de l'instruction par le processeur.
6. Stockage des résultats , puis retour à l'étape 1.

+1

## 4. Performances de l'architecture d'un processeur

Les performances du processeur impactent tout l'ordinateur. Deux processeurs avec la même fréquence peuvent avoir des performances différentes selon leur architecture.

+1

- **Formule simplifiée** :  $P_t = \frac{S}{I_t \cdot C} \cdot \$$ .
- **Composantes de la formule** : **P** (Performance en millions d'instructions par seconde - MIPS), **S** (Fréquence d'horloge), **I** (Nombre d'instructions), **C** (Nombre moyen de cycles par instruction) .
- **Caractéristiques influentes** : La fréquence, la taille du bus de données (8, 16, 32, 64 bits), la technologie et l'architecture interne .

## 5. Architecture Von Neumann

Conçue en 1945, c'est la base de presque tous les ordinateurs actuels .

- Elle repose sur : une mémoire centrale unique (pour données et instructions), une Unité de Commande, une UAL, un accumulateur et des Entrées/Sorties .
- **La limite principale** : Le "goulot d'étranglement de Von Neumann", dû à l'interconnexion (le bus) entre le CPU et la mémoire centrale unique qui limite la vitesse.

## 6. Structure d'un système à microprocesseurs

Un système à microprocesseur s'articule autour de trois bus principaux:

- **Bus de données** : Bidirectionnel, transfère les données (taille : 8, 16, 32, 64 bits).
- **Bus d'adresse** : Unidirectionnel, envoie l'adresse de la mémoire ou du périphérique cible.
- **Bus de contrôle** : Transmet les signaux de commande (ex: Lecture/Écriture, requête mémoire) .

## 7. Opérations d'Entrées/Sorties

Ce sont les méthodes de transfert de données entre le CPU et les périphériques:

+1

- **Par Polling (programme)** : Le CPU interroge continuellement le périphérique. C'est lent et ça monopolise le CPU.

+1

- **Par Interruption** : Utilise un contrôleur (PIC). Le périphérique demande l'accès, le CPU met son programme principal en pause (sauve son contexte), exécute le transfert, puis reprend .
- **Par DMA (Direct Memory Access)** : Utilisé pour les périphériques rapides. Le transfert se fait directement entre la mémoire et le périphérique. Le CPU se déconnecte du bus et n'intervient pas pendant le transfert .

## 8. Critères de qualité d'un algorithme (d'ordonnancement)

L'ordonnancement choisit quel processus accède au processeur. Un bon algorithme doit équilibrer ces critères parfois contradictoires:

+2

- **Efficacité** : Éviter les temps morts du processeur.
- **Temps de réponse** : Réagir rapidement aux événements.
- **Impartialité** : Partage équitable du processeur entre les processus.
- **Débit** : Faire un maximum de tâches dans un temps donné.

## 9 & 14. Programmation Parallèle

Face aux limites physiques d'accélération des processeurs (vitesse de la lumière, dissipation d'énergie), on utilise des processeurs "multi cores" . Pour exploiter ces cœurs, il faut des programmes parallèles.

+1

- **La méthodologie de Ian Foster** pour créer un programme parallèle compte 4 étapes:
  1. **Partitionnement** : Diviser le travail et les données en petites tâches.
  2. **Communication** : Déterminer comment les tâches s'échangent des données.
  3. **Agrégation** : Combiner les petites tâches en tâches plus grandes.
  4. **Mapping (Assignment)** : Assigner les tâches aux processus/threads.
- **Coordination** : Elle exige le *load balancing* (équilibrer les charges de travail), la *synchronisation* (gérer l'ordre dans le temps) et la *communication* entre processus .

## 11. Architecture MIMD (Multiple Instruction, Multiple Data)

Les processeurs exécutent des programmes différents sur des données différentes. Ses variantes :

+2

- **Centralisée** : Les processeurs partagent une mémoire commune.
- **Horizontale** : Chaque processeur a un rôle défini (ex: CPU central ou spécialisé E/S).
- **Hybride** : Chaque processeur a une mémoire locale pour soulager le bus central.
- **À bus hiérarchisé** : Utilise des sous-bus pour décharger le bus principal.

## 12. Systèmes à mémoire distribuée

Dans ce système, chaque processeur est couplé à sa propre mémoire privée (ex: un Cluster de PC connectés par Ethernet). Ils communiquent explicitement en s'envoyant des messages via un réseau (grâce à des interfaces comme MPI, utilisant des fonctions *send* et *receive*).

+3

## 13. Réseaux d'interconnexion

Ils sont vitaux car un mauvais réseau dégrade la performance globale. On les évalue par la **latence** (délai avant le premier octet) et la **bande passante** (vitesse de transfert) .

+1

- **Pour mémoire partagée** : On utilise le *bus* (partagé, mais risque d'encombrement) ou le *commutateur crossbar* (qui utilise des switchs pour router les données) .

+1

- **Pour mémoire distribuée** : L'interconnexion peut être *directe* (switchs connectés directement aux processeurs, ex: architecture en anneau) ou *indirecte* .

## 15. Les "Super machines" et les hautes performances

L'amélioration de l'architecture (Pipe-line, processeurs vectoriels, systèmes parallèles) vise à créer des machines capables de très hautes performances.

+2

- **Applications visées** : Modélisation climatique, recherche sur les protéines (maladies), découverte de médicaments, analyse du *Big Data* .
- **Exemples historiques/actuels** : L'ILLIAC IV (pionnier des processeurs tableaux/SIMD) et les Clusters modernes (systèmes distribués).

+1

---