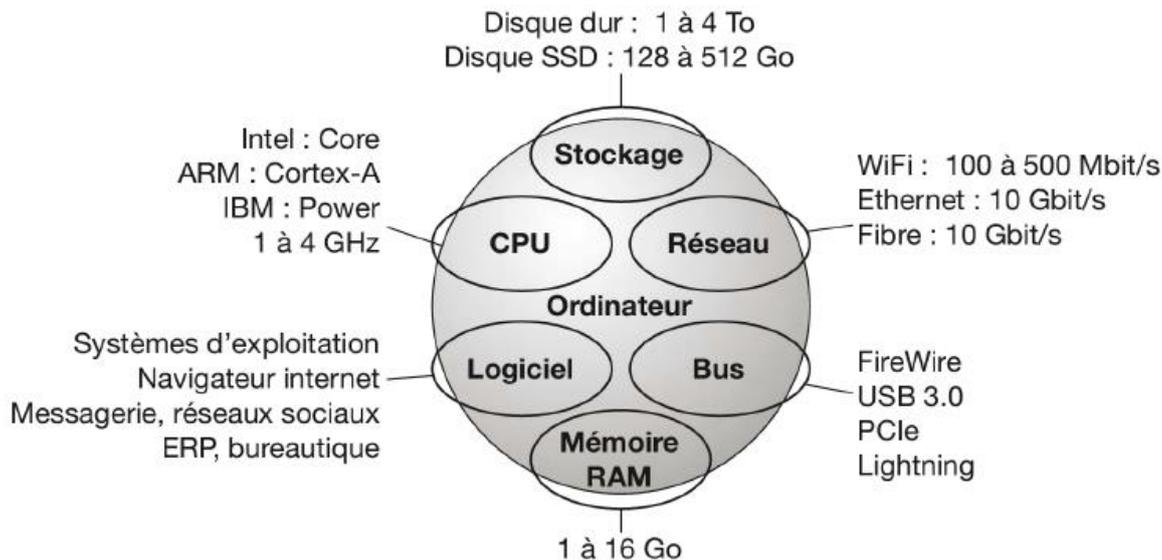
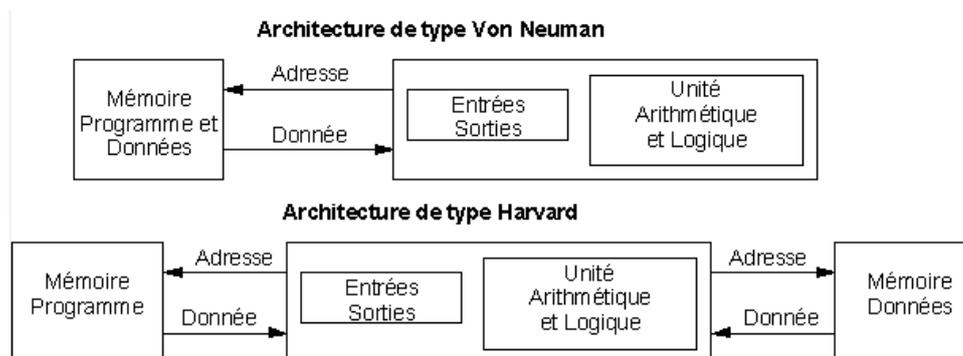


Composants d'un Système Informatique

1. **Les ressources matérielles** (CPU, mémoire, Périphériques d' E/S)
2. **Système d'Exploitation** : Interface entre le matériel et les programmes des utilisateurs
3. **Programmes Applicatifs** (bases de données, jeux vidéos, programmes de travail)
4. **Utilisateurs** Tout ce qui peut lancer un programme (personnes, machines, ...)



L'architecture d'un ordinateur.



Architecture Von Neumann	Architecture Harvard
La même adresse de mémoire physique est utilisée pour les instructions et les données.	Une adresse de mémoire physique distincte est utilisée pour les instructions et les données
Il existe un bus commun pour le transfert de données et d'instructions.	Des bus séparés sont utilisés pour transférer les données et les instructions.
Deux cycles d'horloge sont nécessaires pour exécuter une seule instruction. CISC	Une instruction est exécutée en un seul cycle.(RISC)
C'est moins cher en coût.	C'est plus cher que Von Neumann Architecture.
Le processeur ne peut pas accéder aux instructions et lire/écrire en même temps.	Le processeur peut accéder aux instructions et lire/écrire en même temps.
Il est utilisé dans les ordinateurs personnels	Utilisé dans les microcontrôleurs et le traitement du signal.

Éléments principaux d'un ordinateur

Le temps : L'horloge temps réel ou Real-time Clock (RTC) :

Processeur central : Aussi appelé CPU (Central Processing Unit), Il contient

1. **UAL** : Unité Arithmétique (addition, soustraction,...) et Logique (ET, OU,...)
2. **Unité de commande** : Coordinateur général, Lit les instructions et les exécutent
3. **Les registres** : Mémoire à accès rapide utilisée pour les calculs
4. **Mémoire cache** : Mémoire intermédiaire pour optimiser les performances.

Mémoire centrale : Stocke les programmes et les données, Enregistre les résultats,

Unités d'entrées / sorties, En entrée (clavier, souris, disque dur,...) et sortie (carte graphique, disque dur,...)

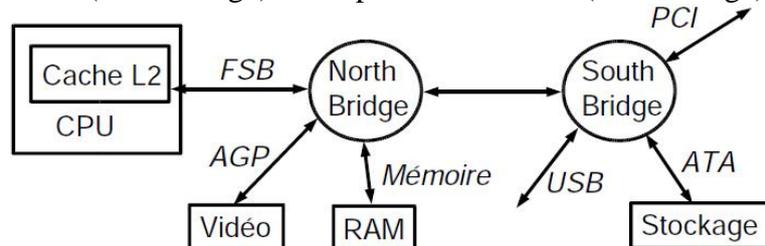
Mémoire : Mémoire cache, Principale RAM et mémoire auxiliaire Disques Magnétiques.....

Bus : Canal partagé (multiplexage). plusieurs bus +/- rapides ou partagés

Dans un PC, bus rapides : Bus système (FSB ou Front Side Bus), Bus de communication avec le CPU, Bus mémoire : communication avec la mémoire, Bus AGP (ou PCI-X) : communique avec carte graphique

Dans un PC, bus plus lents : PCI : cartes réseaux, son ..., Connexion périphérique de stockage (DD, CD, DVD...) avec ATA, SATA, SCSI ..., Connexion de périphériques extérieurs (USB, FireWire..)

Chipset : Composé Pont nord (NorthBridge) bus rapides et Pont sud (SouthBridge):bus lents

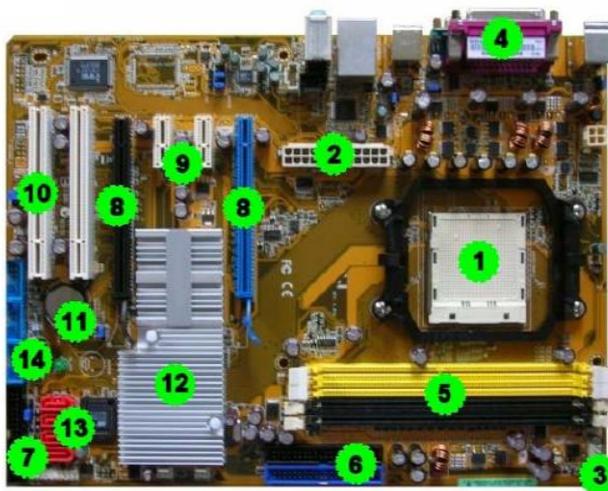


Northbridge : Contrôleur mémoire (rafraîchissement, sélection..),compatibilité limité avec un type de mémoire et un CPU,terminologie: Graphics and Memory Controller Hub (**GMCH**)

Southbridge : dédié aux bus de moindres débits, peut inclure : DMA, RTC, PIC. Terminologie Intel : I/O Controller Hub (**ICH**).

DMA : Direct Memory Access, **FSB** : Front Side Bus, **INT** : Interruption, **IRQ** : Interrupt ReQuest, **ISR** : Interrupt Service Routine, **MMU** : Memory Management Unit, **OS** : Operating System, **PIC** : Programmable Interrupt Controleur, **RAM** : Random Access Memory, **RTC** : Real Time Clock.

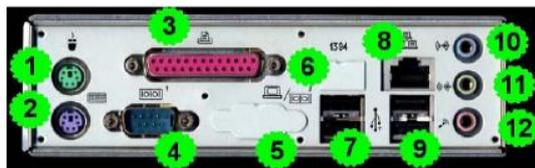
La carte mère

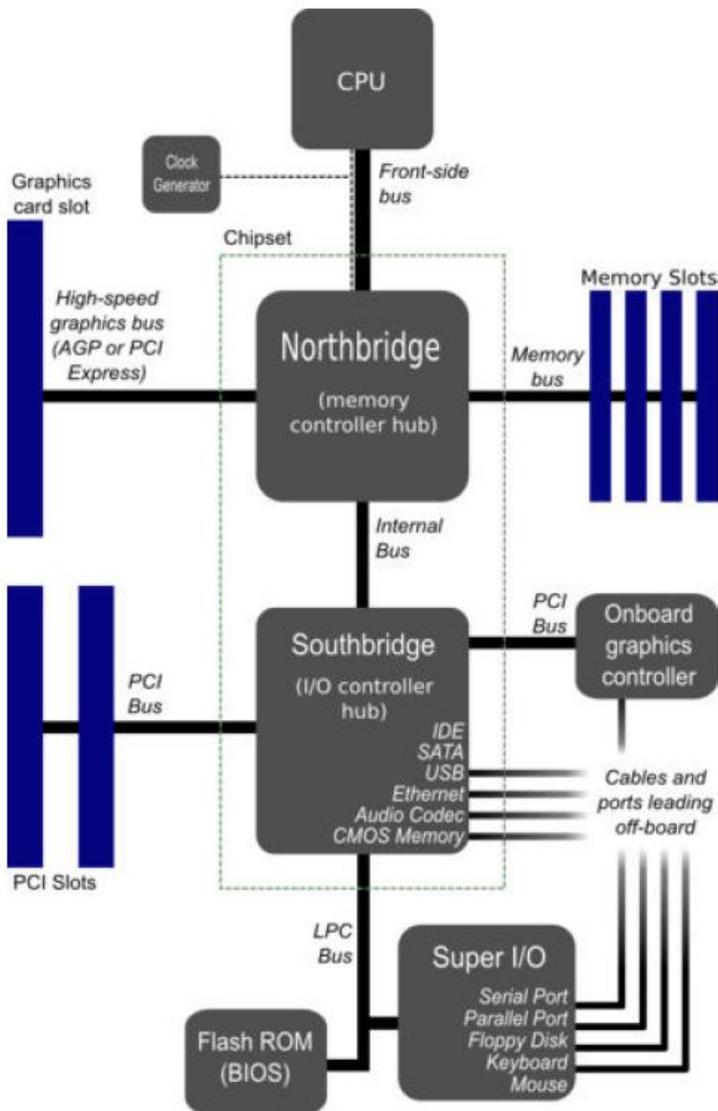


- 1 : Socket ou slot du processeur
- 2 : Prise de l'alimentation électrique de la carte mère (ici du 24 broches)
- 3 : Prise de l'alimentation électrique du ventilateur du processeur
- 4 : ports situés à l'arrière (voir diapositive suivante)
- 5 : Slot(s) mémoire (ici pour DDR2-SDRAM sur 240 broches)
- 6 : Port IDE
- 7 : Port Floppy
- 8 : Port PCI Express 16x
- 9 : Port PCI Express 1x
- 10 : Port PCI
- 11 : Pile
- 12 : Chipset
- 13 : Ports Serial ATA
- 14 : Ports USB interne

Les ports de la carte mère

- 1 : Port PS/2 souris
- 2 : Port PS/2 clavier
- 3 : Port parallèle
- 4 : Port série
- 5 : Port série
- 6 : Port Firewire (l'emplacement est prévue, mais la carte est dépourvue de ce port)
- 7 : Port USB
- 8 : Port LAN
- 9 : Port USB
- 10 : Prise Jack d'entrée de son auxiliaire
- 11 : Prise Jack de sortie (pour un branchement sur haut-parleurs ou un casque)
- 12 : Prise Jack d'entrée de son microphone



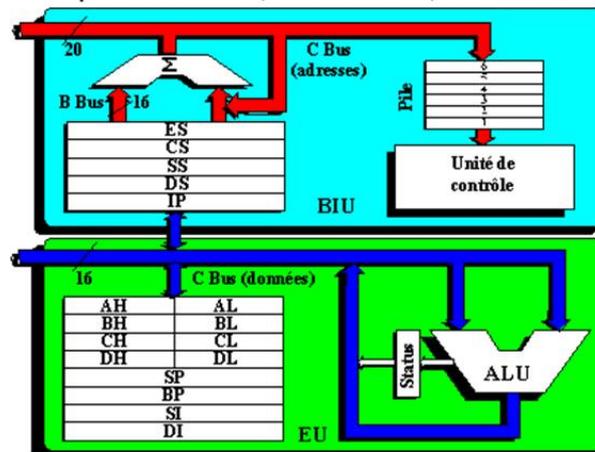


Mémoire : Mémoire cache, Principale et Disques Magnétiques (deux modes d'adressage réel et protégé)

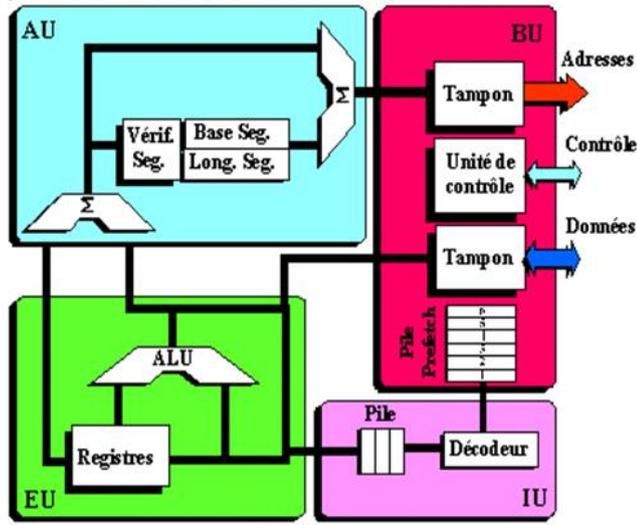
Le mode réel (physique) : Le mode d'exécution du 8086 (processeur 16 bits) avec bus d'adresse de 2^{20} bits (1Mo), des registres de 16 bits permettent d'adresser une mémoire linéaire de 2^{16} bits (64 Ko)

- pour accéder au 1 Mo de mémoire : il utilise un registre *segment* de 16bits et un offset : l'adresse mémoire sur 16bits. Alors l'adresse finale= $segment \times 16 + offset$
 - Un segments adressables de 64 Ko commençant à l'adresse de base ($segment \times 16$)
- Le processeur utilise les registre segments CS, DS,ES, SS

μP 16 bits (Intel 1978) : **8086**

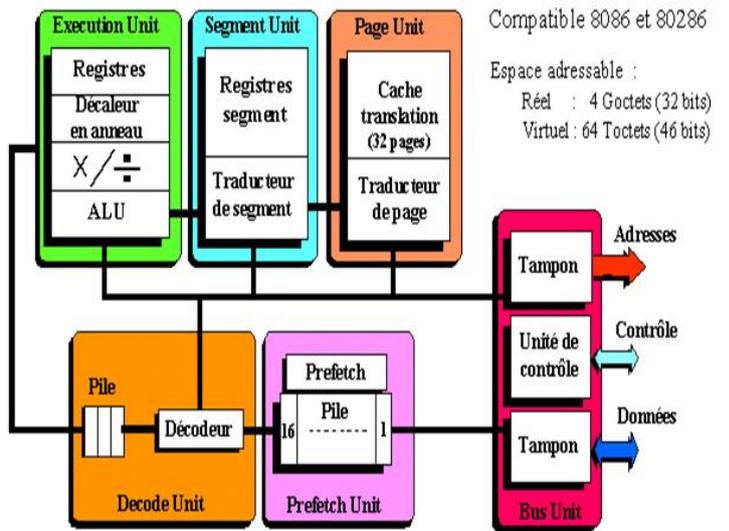


μP 16 bits (Intel 1982) : 80286



Avec 80286 (16 bits, bus adresse de 24 bits)

μP 32 bits (Intel 1985) : 80386

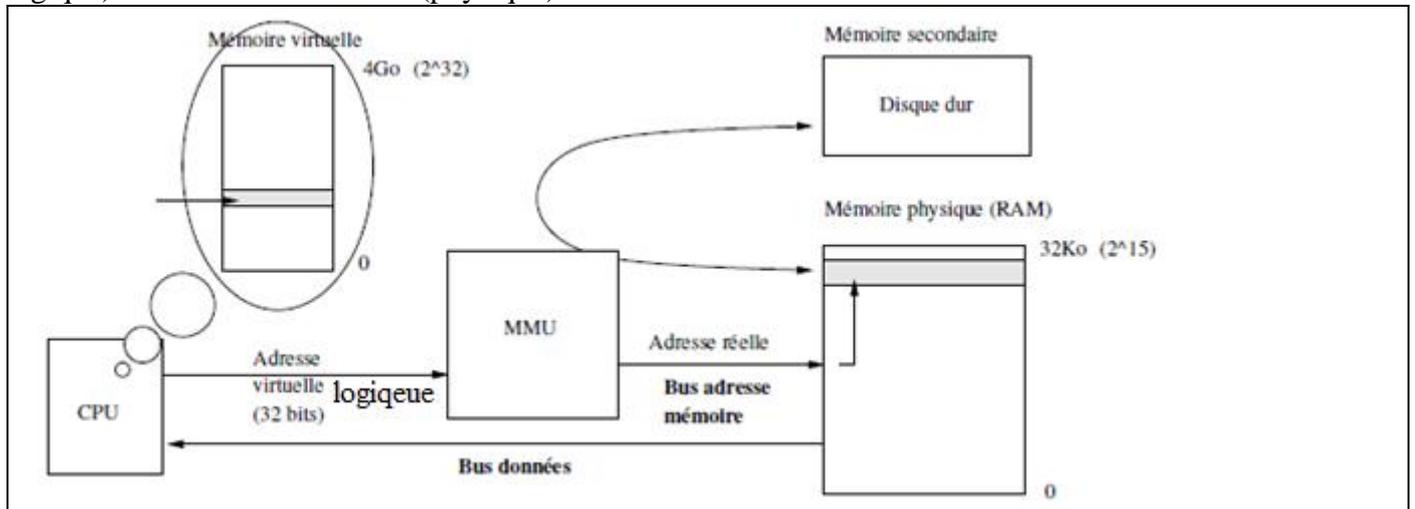


Compatible 8086 et 80286
Espace adressable :
Réal : 4 Goctets (32 bits)
Virtuel : 64 Toctets (46 bits)

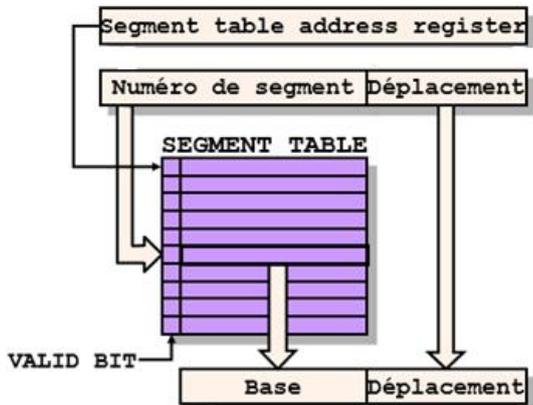
Le mode protégé segmentation et pagination

Adressage de la mémoire

Unité de Gestion de la mémoire MMU (Memory Management Unite: Translation d'adresses virtuelles (ou logique) vers une adresse réelle (physique)).

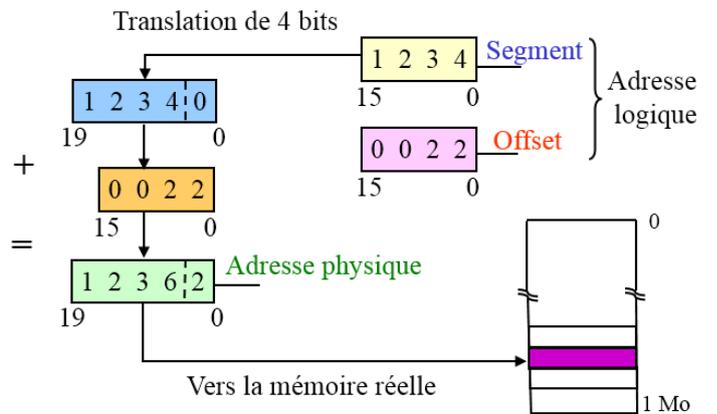


Segmentation

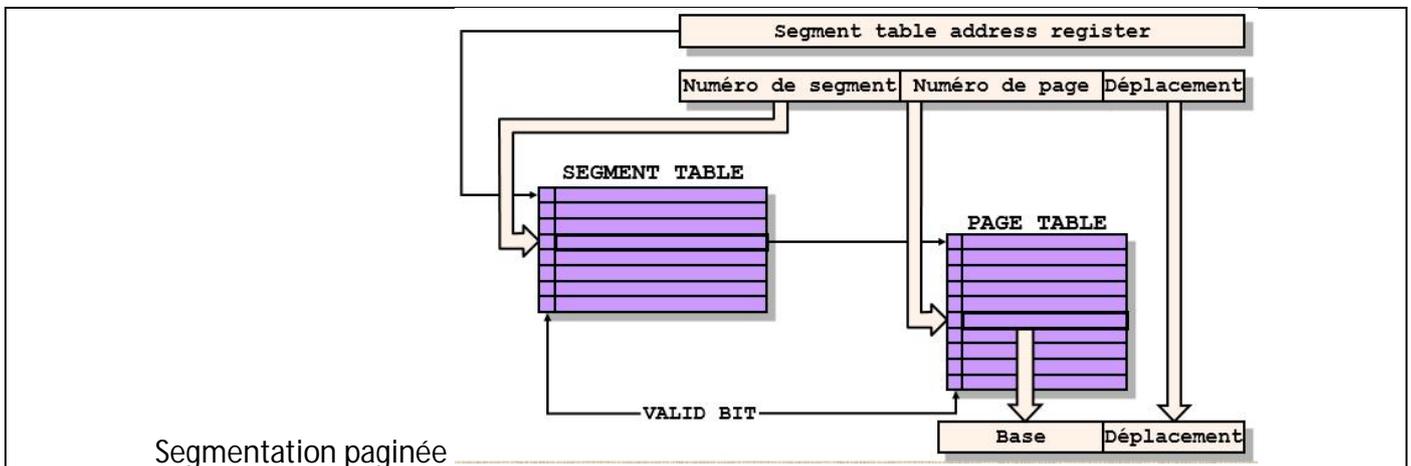


Adressage réel

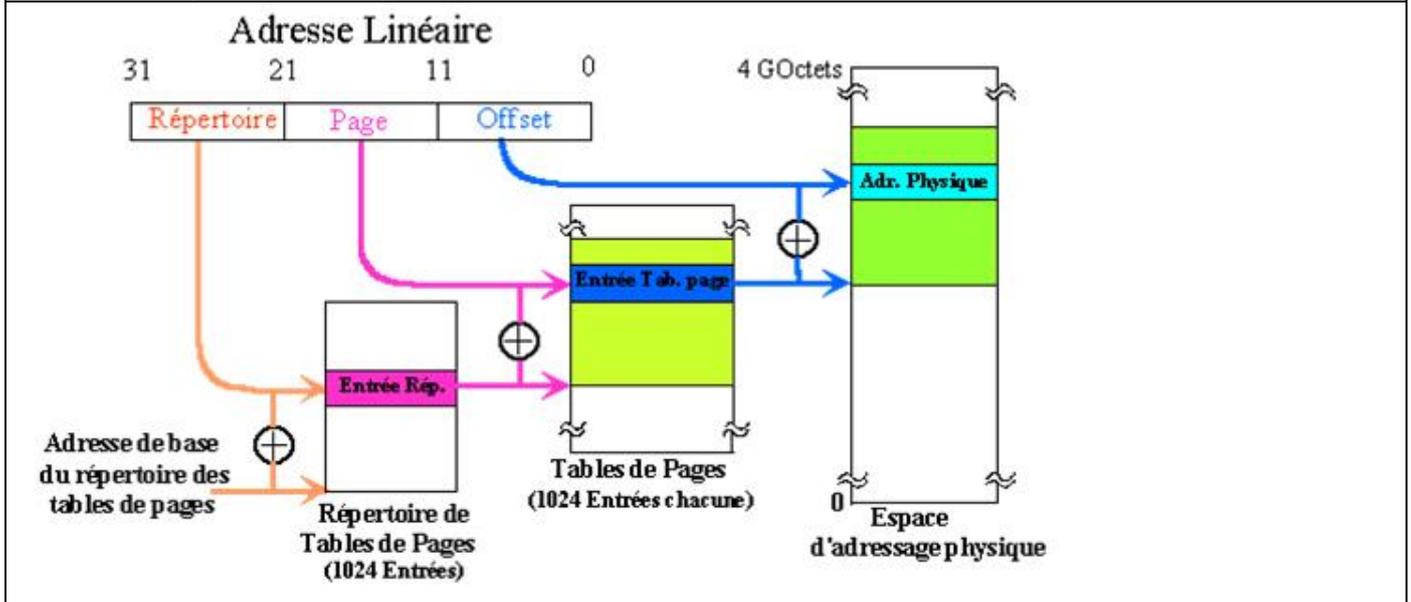
Génération d'une adresse physique



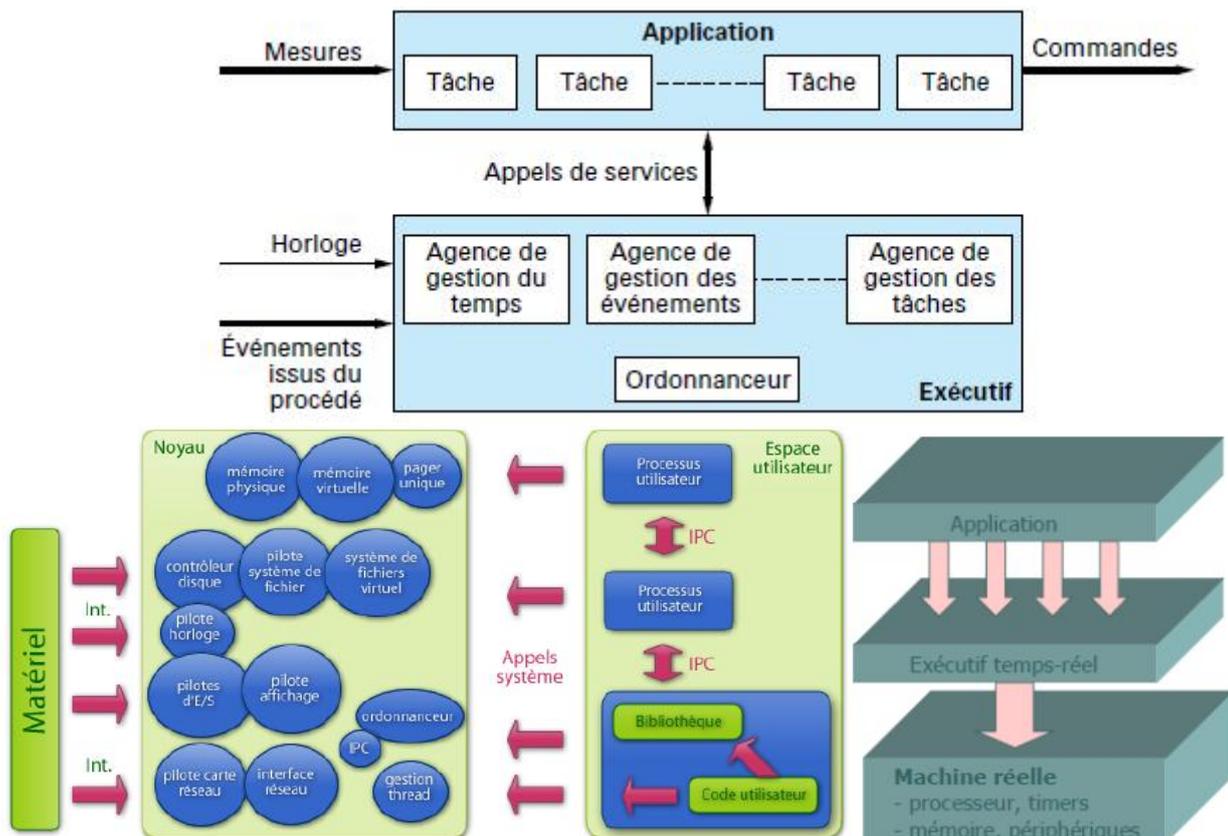
Mode 8086 offset court 16 bits

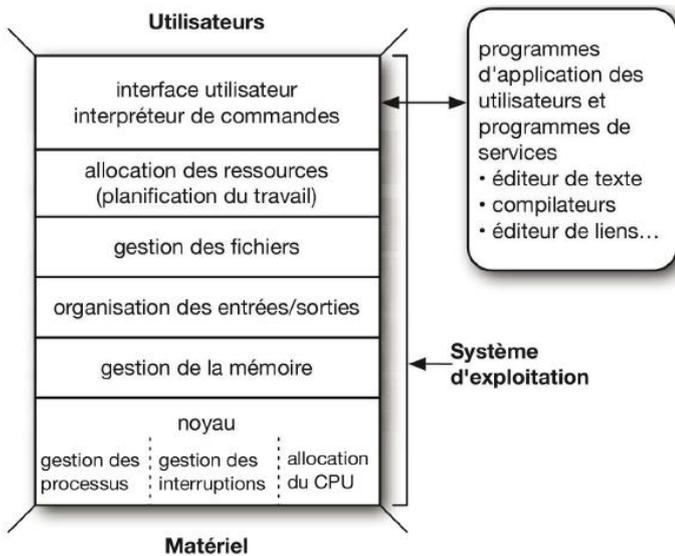


Segmentation paginée

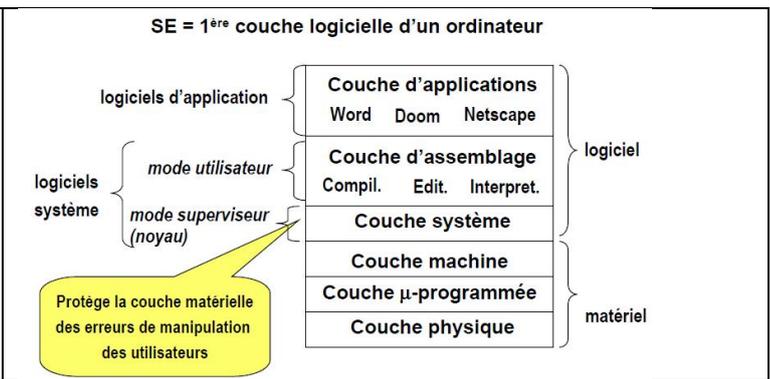


Système d'exploitation : Ensemble de programmes responsables de la liaison entre les ressources matérielles d'un ordinateur et les applications informatiques de l'utilisateur (traitement de texte, jeux vidéo, etc). Il fournit aux programmes d'applications des points d'entrée génériques pour les périphériques.

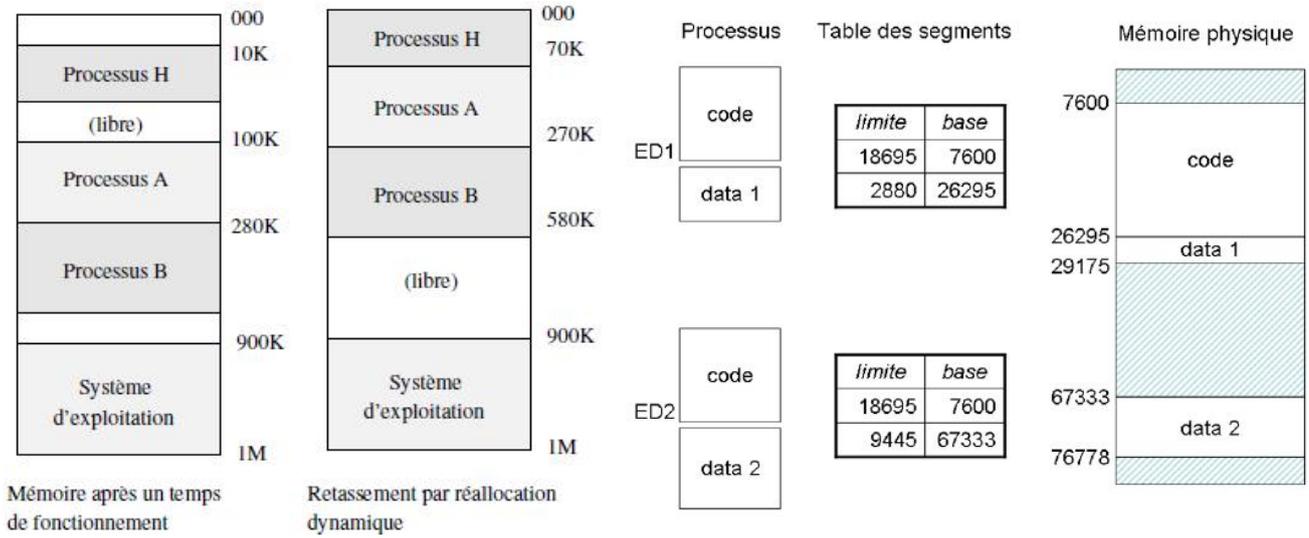




- Types de services fournis
- Gestion des **tâches**
 - Gestion des **synchronisations**
 - Gestion des **communications**
 - Gestion de la **mémoire**
 - Gestion du **temps**
- Gestion des **périphériques**



Gestion des processus (taches)



Un **programme** spécifie l'exécution séquentielle d'une liste d'instructions par un agent dit **processeur**. On appelle **processus** cette exécution.

Bloc de contrôle de processus (Process Control Block (PCB))

- tâche élue (exécution) : un processeur est alloué à la tâche
- tâche bloquée : la tâche est en attente d'une ressource ou d'une synchronisation
- tâche prête : la tâche est en attente du processeur



Temps d'exécution pire des cas d'un processus

Analyse bas niveau

Exemple

```
somme = 0;
nbpairs = 0;
for (i=0; i < N; i++)
{ somme += tab[i];
if (tab[i] % 2 == 0)
nbpairs++; }
```

Sur un calculateur et sous l'Hypothèses pas de pipeline et chaque instruction dure 1 cycle machine

<pre> mov r0,#0 mov r1,#0 mov r2,#0 adr r10,tab adr r11,N ldr r11,[r11] for: cmp r2,r11 bcs fin ldr r4,[r10],#4 add r0,r0,r4 if: and r3,r4,#1 cmp r3,#0 bne suite add r1,r1,#1 suite: add r2,r2,#1 b for fin: </pre>	<div style="font-size: 2em; color: blue;">}</div> <p>1 × 6 cycles</p> <div style="font-size: 2em; color: blue;">}</div> <p>(N+1) × 2 cycles</p> <div style="font-size: 2em; color: blue;">}</div> <p>N × 5 cycles</p> <div style="font-size: 2em; color: blue;">}</div> <p>n × 1 cycle</p> <div style="font-size: 2em; color: blue;">}</div> <p>N × 2 cycles</p>	<p style="color: blue; font-weight: bold;">$T = 8 + 9 \times N + n$</p> <p>si $N = 5$ et $n = 2$: $T = 55$ 10 chemins possibles</p> <p>si $N = 8$ et $n = 4$: $T = 84$ 70 chemins possibles</p> <p>si $N = 10$ et $n = 10$: $T = 108$</p>
--	--	---

Analyse de flot : Détermine les chemins d'exécutions possibles

Analyse de flot : exemple

```

int f(unsigned int x) {
char i=0;
while (i<100) {
if ( i<20 || x<30 )
x = x + 1;
else
x = x * 2;
i++;
}
return(x)
}

```

- la boucle itère au plus 100 fois
- E ne peut pas être exécuté dans les 20 premières itérations
- D ne peut être exécuté que dans les 30 premières itérations

$$T = \sum_{i,j} x_{ij} c_{j[i]}$$

↑
temps d'exécution d'un chemin

↑
coût du bloc j dans la séquence i-j
nombre d'exécutions de l'arc i-j

La mesure des performances d'un processeurs.

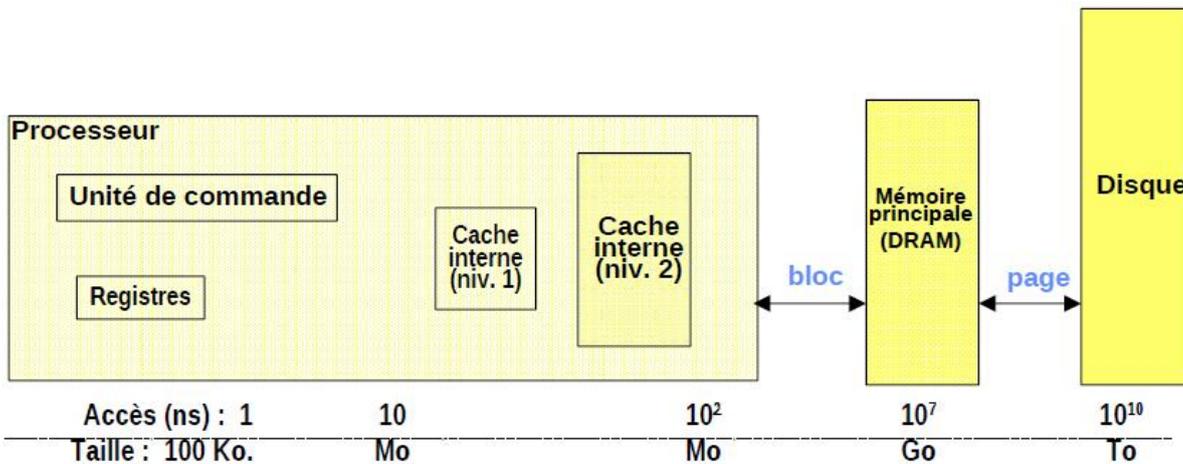
- Pour le processeur : Temps de réponse (d'exécution) = durée s'écoulant entre le début et la terminaison d'une tâche.
- Performances = 1/Temps exécution : Plus le temps d'exécution est court, plus les performances sont grandes.

$$\text{Performances } M_1 > \text{Performances } M_2 \Leftrightarrow \text{Temps } M_1 < \text{Temps } M_2$$

$$M_1 \text{ est } n \text{ fois plus rapide que } M_2 \Leftrightarrow \begin{cases} (\text{Performances } M_1 / \text{Performances } M_2) = n \\ ((1/\text{Temps } M_1) / (1/\text{Temps } M_2)) = n \\ (\text{Temps } M_2 / \text{Temps } M_1) = n \end{cases}$$

Temps moyen d'accès à la mémoire

cache niveau 1 → cache niveau 2 → mémoire → disque



Calcul des performances : Terminologie pour accès à niveau donné de la hiérarchie

- Echec = L'objet désiré n'est pas disponible.
- Succès = L'objet désiré est disponible.
- Taux de succès = % des accès trouvant l'information dans le niveau supérieur
- Taux d'échec = 1 – taux de succès (Nombre d'échecs / Nombre d'accès)
- Temps de succès = temps d'accès à l'information dans le niveau supérieur (Temps d'accès lorsque l'objet est disponible)
- Temps de succès-échec = temps d'accès à l'information dans le niveau supérieur (Temps d'accès lorsque l'objet n'est pas disponible)
- Coût d'un échec ou Pénalité d'échec = (Temps lorsque l'objet n'est pas présent (= temps pour accéder au niveau inférieur + temps pour mettre à jour les informations du niveau accédé).)

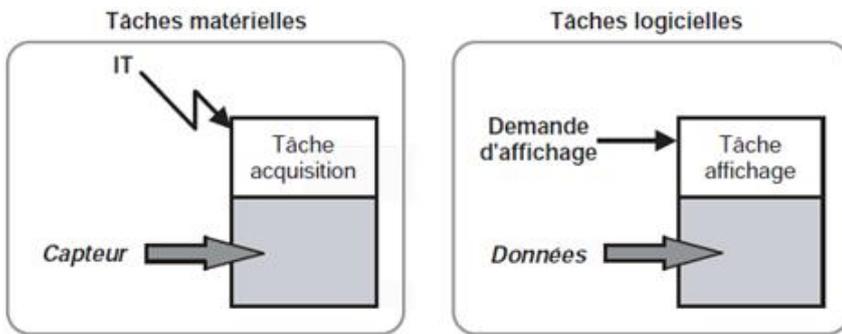
Temps moyen d'accès à une mémoire t_i est donné par la formule suivante

$$t_i = \text{temps}_{\text{accès-succès}} \times \text{taux}_{\text{succès}} + (\text{temps}_{\text{accès-échec}} + t_{i+1}) \times \text{taux}_{\text{échec}}$$

→ Bonne gestion de la mémoire : Temps succès \ll Pénalité d'échec

Ordonnement

Les systèmes industriels fonctionnant en temps réel, doivent délivrer des résultats exacts dans des délais imposés. Ces deux besoins (résultats et délais) et leurs améliorations fait appel à plusieurs domaines (système temps réel, algorithme ordonnancement, architecture processeur, vitesse d'exécution...)



Classification du Temps Réel

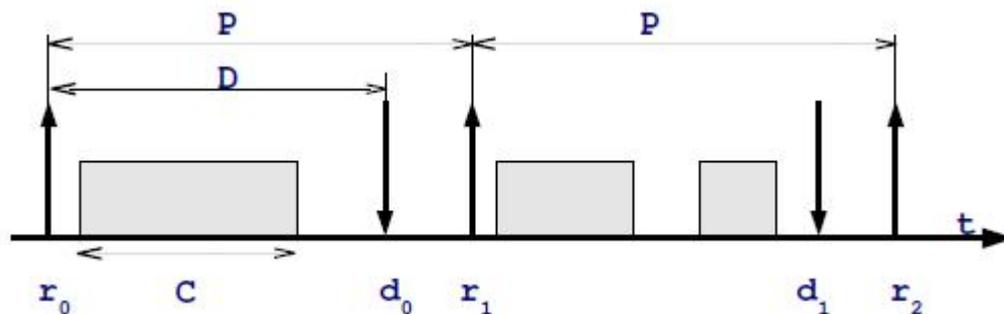
Hard Real Time (Temps réel critique): Les contraintes (échéances) doivent être respectées. Exemple : systèmes industriels (production), systèmes embarqués (avionique, automobile...),...

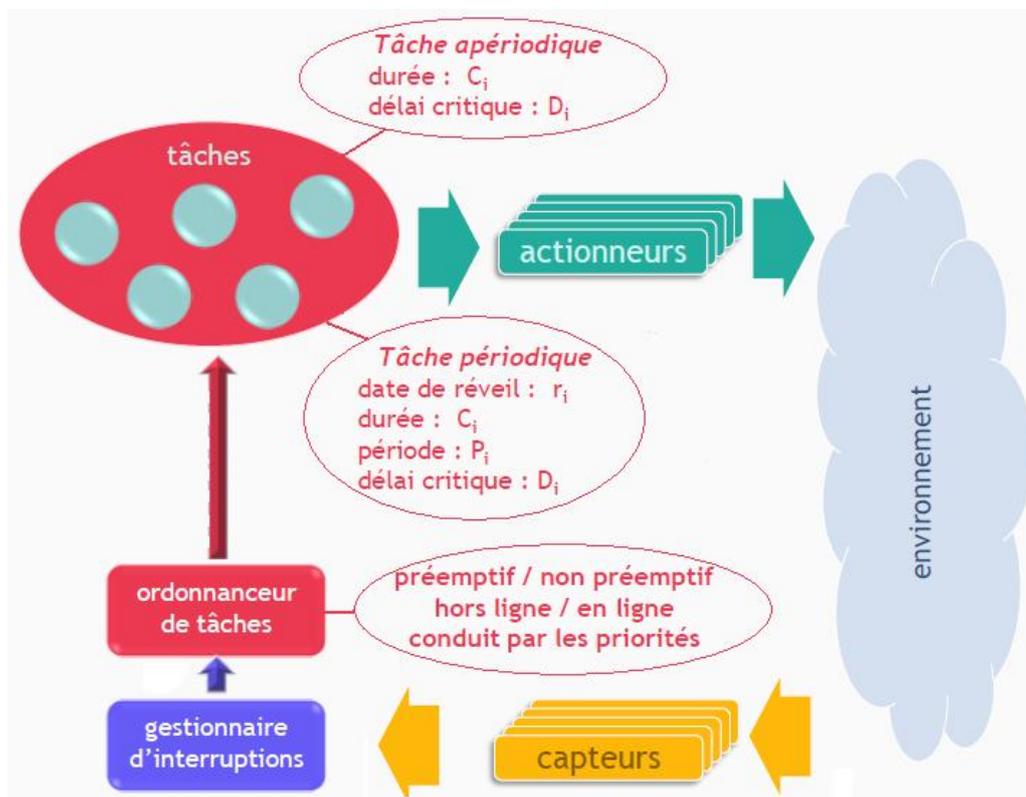
Soft Real Time (Temps réel à contrainte souple): Mieux respecter les contraintes. Exemple : les applications de vidéo-conférence, ...

- Une tâche temps réel soumise à des contraintes de temps, plus ou moins strictes (instant de démarrage, instant de fin, temps absolus ou relatifs à d'autres tâches)
- Le but de l'ordonnancement est de permettre le respect de ces contraintes, il doit permettre de borner les effets d'incidents ou de surcharges

Caractéristiques des tâches

- r : date de réveil (moment du déclenchement de la 1^{ère} requête d'exécution)
- C : durée d'exécution maximale (capacité)
- D : délai critique (délai maximum acceptable pour son exécution)
- P : période (si tâche périodique)
- $d = r + D$: échéance (si tâche à contraintes strictes)





Définitions pour l'ordonnancement

Configuration : ensemble de n tâches mises en jeu par l'application (départs échelonnés ou simultanés)

$$\text{Facteur d'utilisation du processeur } U = \sum_{i=1}^n \frac{C_i}{P_i}$$

$$\text{Facteur de charge du processeur } CH = \sum_{i=1}^n \frac{C_i}{D_i}$$

Intervalle d'étude : intervalle de temps minimum pour prouver l'ordonnançabilité d'une configuration le PPCM des périodes dans le cas d'une configuration de tâches périodiques

$$T = \text{PPCM}\{P_1, \dots, P_i, \dots, P_n\}$$

Ordonnancement Rate Monotonic (RM)

la tâche de plus petite période est la tâche la plus prioritaire

Ordonnancement Inverse Deadline (ID)

la tâche de plus petit délai critique est la tâche la plus prioritaire.

Ordonnancement Earliest Deadline (ED)

La priorité maximale est accordée à la tâche dont l'échéance est la plus proche

Ordonnancement Least Laxit (LL)

la tâche de plus petite marge est la tâche la plus prioritaire

$$\text{marge} = \text{échéance} - \text{temps de calcul restant} - \text{temps courant}$$