

FAMILLES DE PROCESSEURS

■ CISC (Complex Instruction Set Computers)

- Instructions complexes
- Les instructions lisent leurs opérantes en mémoire
- Type des processeurs jusqu'aux années 80
- ex: IBM 360-370-390, DEC VAX, Motorola 680x0, Intel x86,

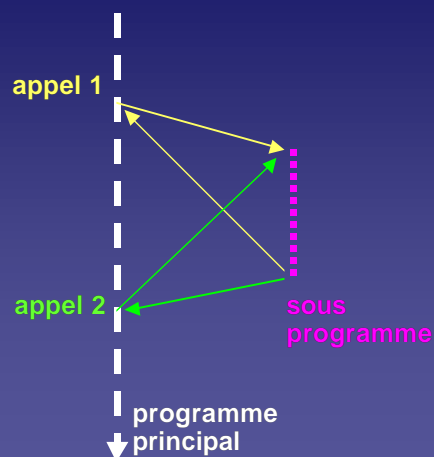
■ RISC (Reduced Instruction Set Computers)

- Instructions simples (et rapides)
- Les instructions lisent leurs opérantes dans des registres
- Type des processeurs des années 80
- Programmes plus longs
- ex: IBM-Motorola PPC, SUN SPARC, DEC ALPHA,

NOTION DE SOUS-PROGRAMME

■ La notion de sous-programme correspond à la réutilisation de **séquences communes** entre plusieurs parties d'un programme

■ Elle correspond aussi à la constitution d'**outils logiciels** qui facilitent, et clarifient la programmation



REGISTRE D'ETAT

- Il existe un registre particulier, qui stocke l'état du processeur.
- Ce registre contient des bits qui décrivent les propriétés du dernier résultat calculé.
 - Signe, nullité, débordement, parité

SOUS-PROGRAMME

- Un sous-programme doit pouvoir être appelé de plusieurs endroits différents.
- L'exécution du programme appelant doit reprendre, juste après l'appel, lorsque le sous-programme se termine.
- Il existe des instructions spéciales d'appel de sous-programme (appelées **CALL**)
- Il existe une instruction spéciale, appelée **RET**, pour revenir au programme appelant après l'exécution d'un sous-programme.

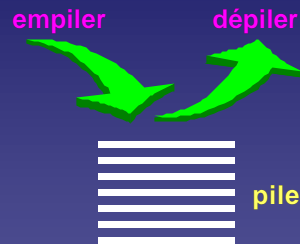
NOTION DE PILE

■ Une pile est le rangement d'objets les uns sur les autres. (par exemple: une pile de papiers)

■ On peut:

- empiler (un papier)
- dépiler (un papier)

■ On dépile les objets dans l'ordre inverse dans lequel on les a empilés.
Les premiers empilés sont les derniers dépilés!



APPEL, ET RETOUR, DE SOUS-PROGRAMMES

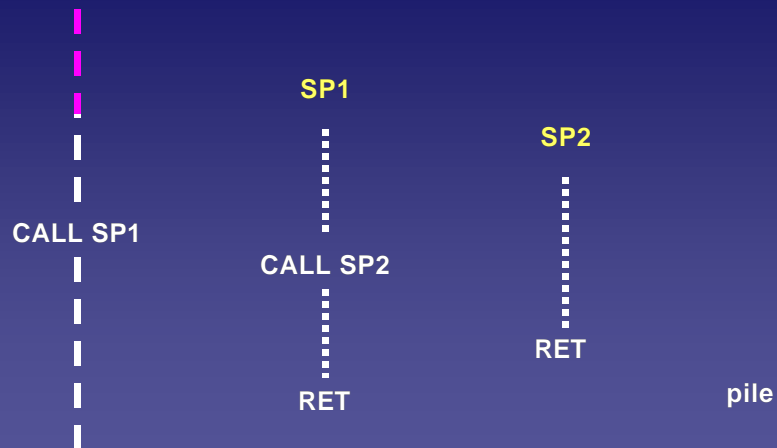
■ On utilise une pile pour gérer les appels, et les retours, de sous-programme.

■ A l'appel d'un sous-programme:

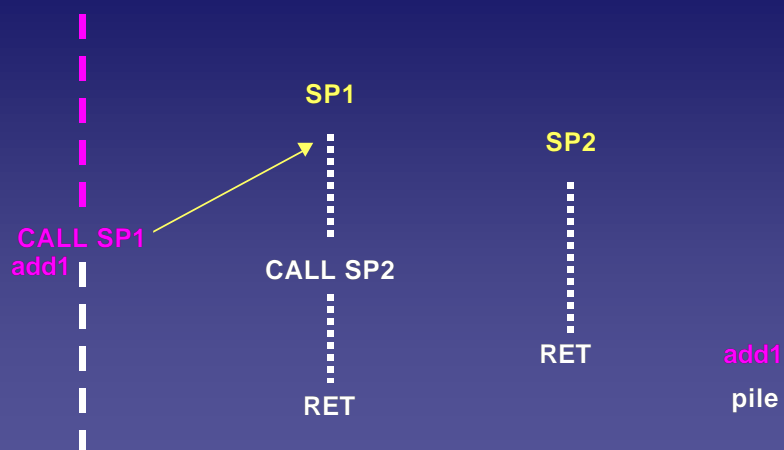
- l'instruction CALL empile l'adresse de l'instruction qui suit le CALL
- à la fin d'un sous-programme, l'instruction RET dépile l'adresse de retour et s'y branche.

■ Cette technique permet d'avoir plusieurs appels imbriqués de sous-programmes.

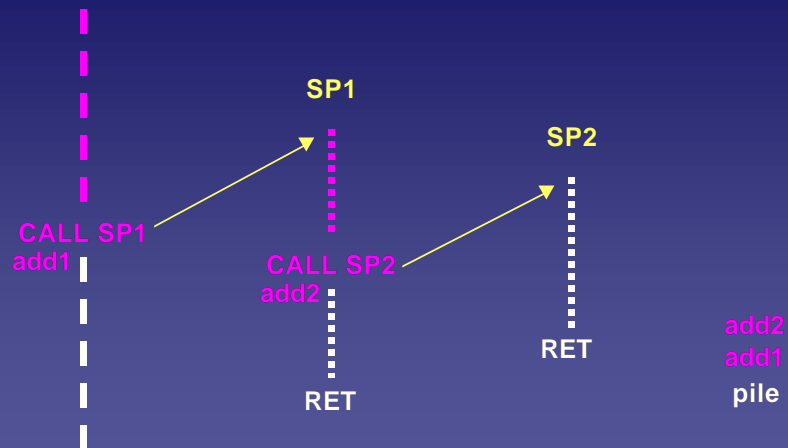
APPEL D'UN SOUS-PROGRAMME



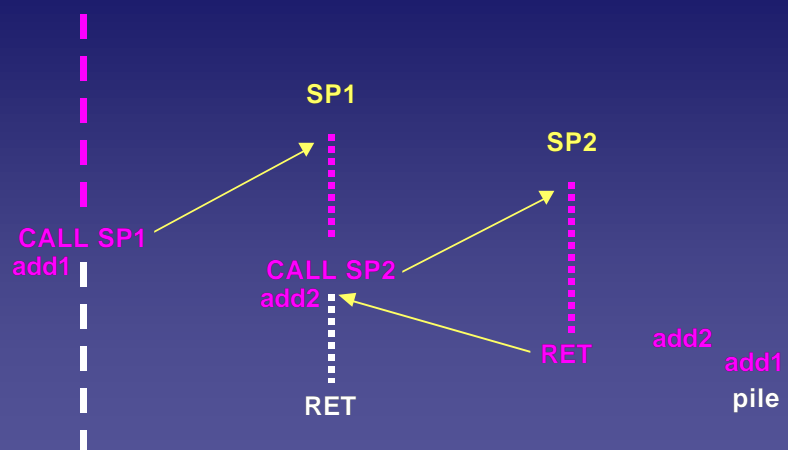
APPEL D'UN SOUS-PROGRAMME



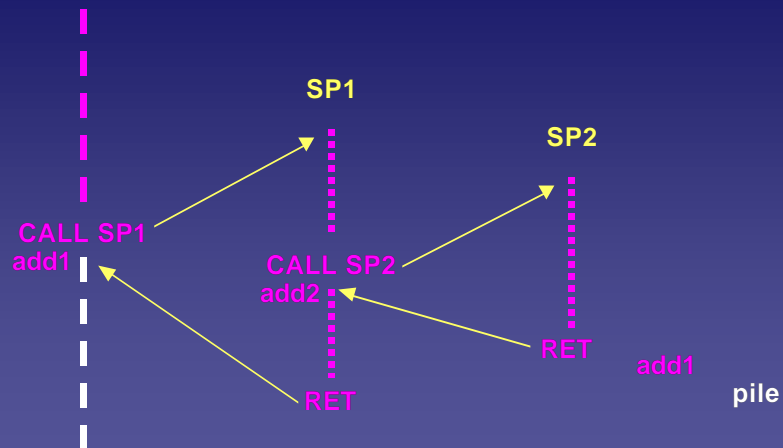
APPEL D'UN SOUS-PROGRAMME



APPEL D'UN SOUS-PROGRAMME



APPEL D'UN SOUS-PROGRAMME



PERTURBATIONS DE L'EXECUTION

■ Le fonctionnement d'un processeur peut être perturbé par:

- Des erreurs dans le processus d'exécution
- Des requêtes extérieures

ERREURS D'EXECUTION

■ Des erreurs peuvent apparaître dans le processus d'exécution

par exemple:

- Des codes opération invalides
- Des divisions par zéro
- Des débordements arithmétiques
-

■ Le processeur ne sait plus quoi faire.

Il lui faut un mécanisme pour demander de l'aide:

- Cette "aide" est spécifique du travail en cours => elle doit être prévue par l'utilisateur

REACTIONS A UNE ERREUR

■ Exemple: Occurrence d'un débordement arithmétique Peut être:

- Une saturation => affecter la valeur maximum
- Une erreur dans la logique du programme => tout arrêter
- Un travail sur des nombres en longueur multiple.....

■ Exemple: Occurrence d'un code opération invalide Peut être:

- Une erreur d'exécution (on cherche à exécuter des données)
=> arrêt immédiat
- Une extension logicielle du jeu d'instruction => on lance un sous-programme

REQUETES EXTERNES

■ Le milieu extérieur peut solliciter du travail à l'ordinateur de manière impromptue

Par exemple:

- Appui sur une touche du clavier
- Un top d'une horloge externe => pour compter le temps
- La lecture d'une info sur le disque
-

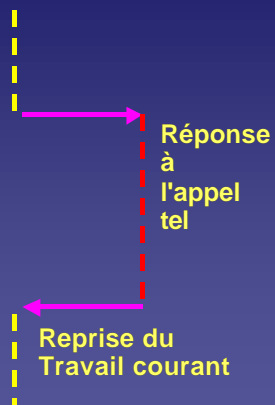
■ Cette requête:

- N'a rien à voir avec ce que fait précisément l'ordinateur au moment de son occurrence
- Son traitement peut être légèrement retardé.
- Elle peut être momentanément ignorée si le travail courant est critique

REQUETE EXTERNE



Travail courant



MECANISME D'INTERRUPTION

■ Mécanisme commun pour le traitement des erreurs d'exécution et des requêtes externes

■ Principe:

- Chaque erreur d'exécution ou requête externe provoque l'appel impromptu du sous-programme associé.
(celui-ci n'a souvent rien à voir avec le programme en cours!)

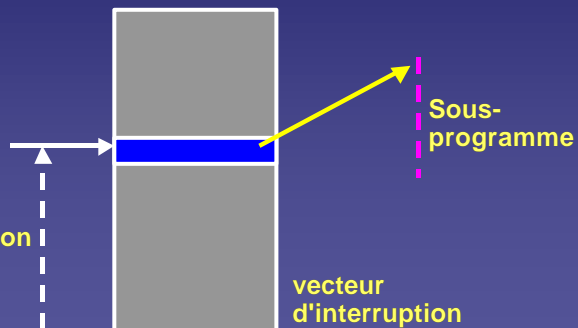
■ C'est un mécanisme très important

EXEMPLE DE MECANISME D'INTERRUPTION

■ Cas des processeurs X86 (ex: Pentium, Athlon)

L'occurrence d'une erreur ou d'une requête externe génère un numéro d'interruption

n° de l'interruption



PRISE EN COMPTE DE L'INTERRUPTION

- **Le lancement du sous-programme impromptu suspend l'exécution en cours => d'où le nom d'interruption**
- **La prise en compte d'une interruption se fait:**
 - Immédiatement s'il s'agit d'une erreur d'exécution
 - Au moment de la prochaine instruction s'il s'agit d'une requête externe **SAUF:**
 - Si le processeur ne souhaite pas être dérangé (travail critique)

MASQUAGE DES INTERRUPTIONS EXTERNES

- **Lorsque l'on ne souhaite pas être perturbé par les interruptions externes on peut les masquer en positionnant un bit dans le registre d'état du processeur.**
 - Bit de **masquage** des interruptions
- **La cause reste en attente jusqu'à ce que ce bit soit remis dans un état qui permette la prise en compte des interruptions.**

ELEMENTS DE MEMORISATION

■ Un ordinateur contient un grand nombre d'éléments de mémorisation

– Processeur:

- Registres de travail
- Registres accessibles par programmation

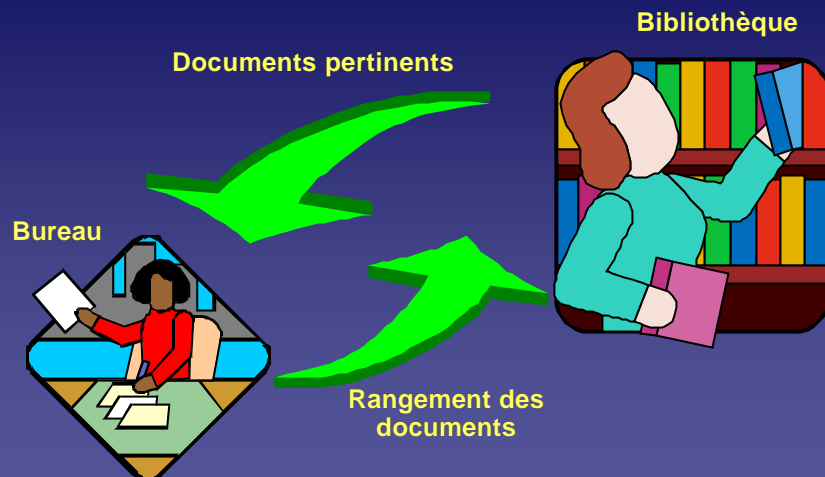
– Ordinateur:

- Mémoire centrale
- Disques
-

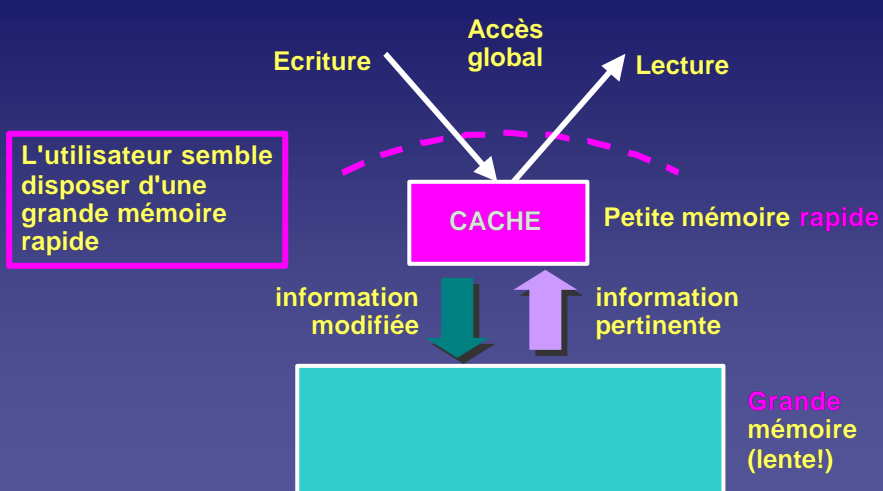
HIERARCHIE DES ELEMENTS DE MEMORISATION

- Plus un élément de mémorisation est grand, plus il est lent (et réciproquement)
- Les plus rapides contiennent les informations les plus utiles à un instant donné.
- Les plus grands stockent la totalité de l'information
- L'info utile migre vers les plus rapides
- L'info modifiée descend mettre à jours les plus vastes

BUREAU <-> BIBLIOTHEQUE



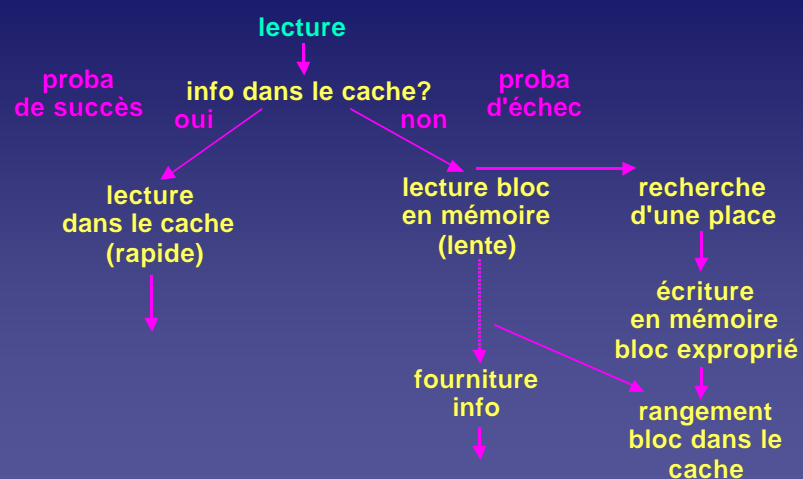
NOTION DE CACHE



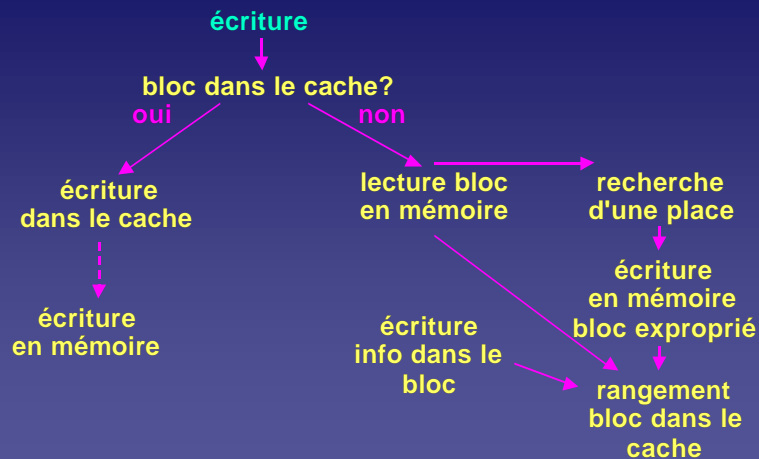
TAILLE DE L'INFO

- Dans le cache l'info est gérée par blocs (64 octets)
- C'est la taille des échanges avec la mémoire
- Le processeur accède le cache par mots

FONCTIONNEMENT DU CACHE (lecture)



FONCTIONNEMENT DU CACHE (lecture)



APPLICATION

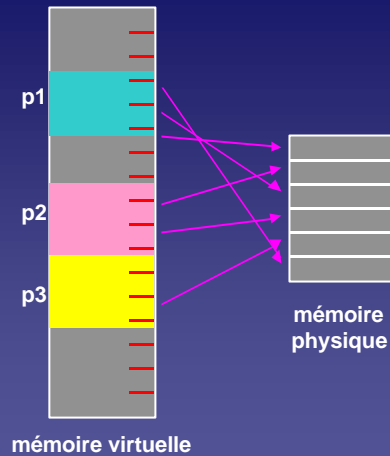
■ Entre le processeur très rapide (temps de cycle de 0,5ns) et la mémoire centrale (100ns), il faut plusieurs niveaux de cache

- Caches L1 dans le processeur (32Ko, 2ns)
 - Cache programme
 - Cache données
- Caches L2 dans le boîtier (512Ko, 10ns)
 - Commun programme et données
- Caches L3 sur la carte-fille.....

MEMOIRE VIRTUELLE

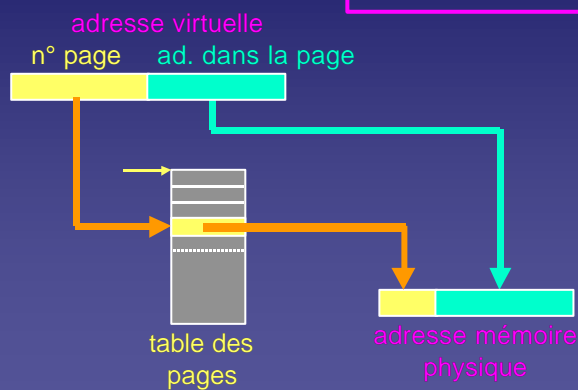
■ La mémoire centrale peut elle-même être vue comme un cache du disque

- Mémoire virtuelle (très grande 4Goctets) sans existence physique => virtuelle
- Transferts par blocs de 4Koctets (pages).



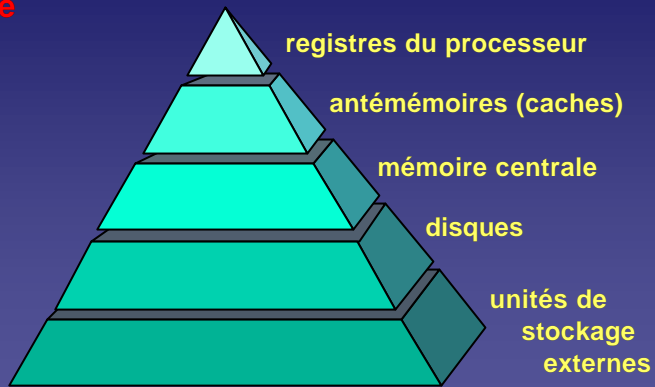
MECANISME DE PAGINATION

Le processeur travaille avec des adresses virtuelles



HIERARCHIE DES ELEMENTS DE MEMORISATION

taille

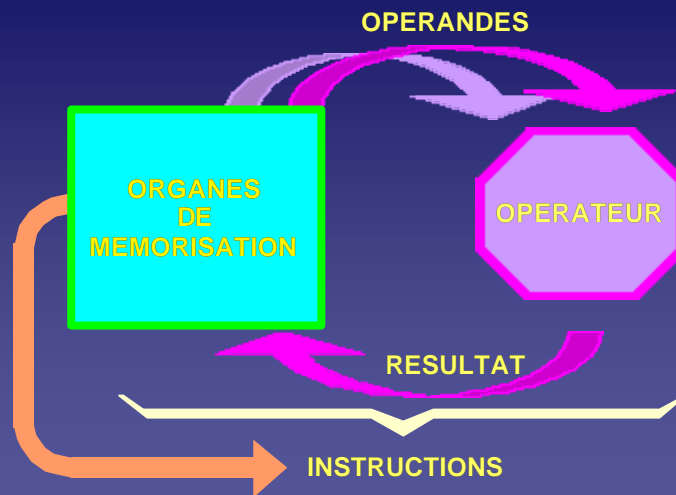


HIERARCHIE DES ELEMENTS DE MEMORISATION

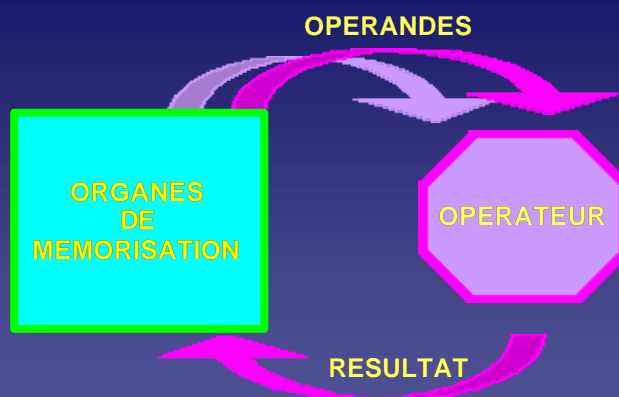
vitesse



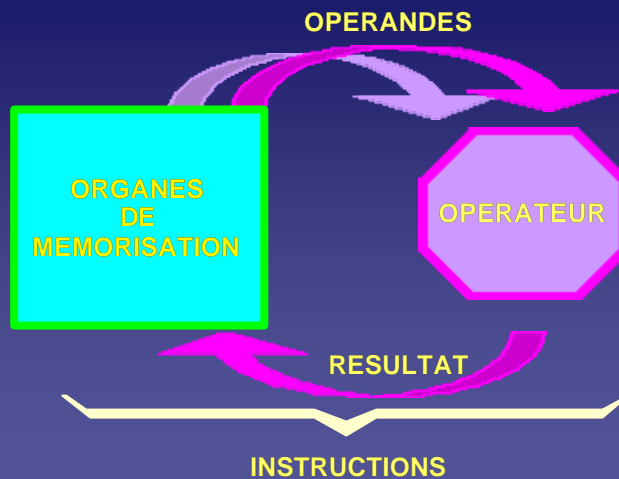
CYCLE ELEMENTAIRE DE CALCUL



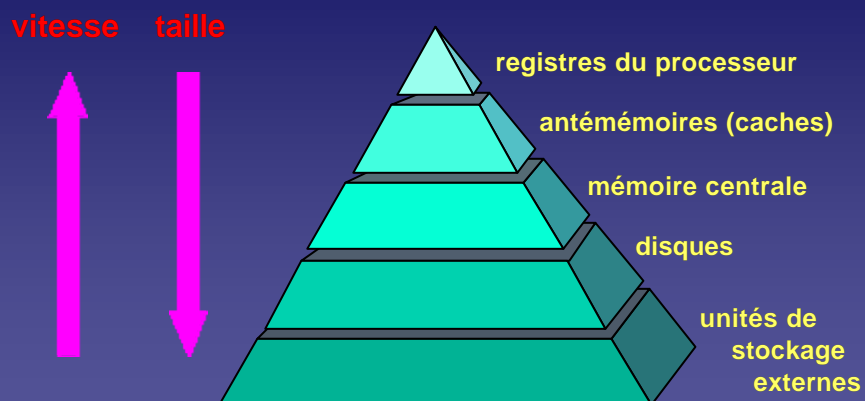
CYCLE ELEMENTAIRE DE CALCUL



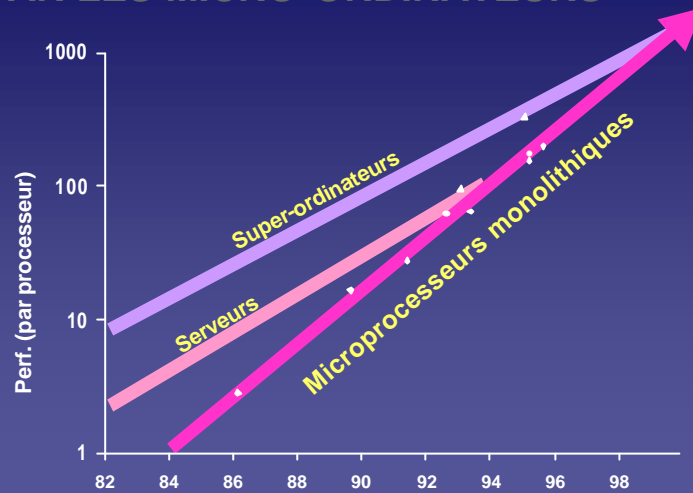
CYCLE ELEMENTAIRE DE CALCUL



HIERARCHIE DES ELEMENTS DE MEMORISATION



ABSORPTION DES GAMMES D'ORDINATEURS PAR LES MICRO-ORDINATEURS



© F. Anceau, janvier 2002, Page 37

PREHISTOIRE

ALGORITHMES

Al Khwarizmi
(perse IXème)

Programmes
Ada de Lovelace
(~1840)

CALCUL MECANIQUE

Machine 4 op.
Schickard (1623)

Pascaline
Pascal (1642)

Machine multiplicatrice
Leibniz (1694)

PROGRAMMATION

Carillons automatiques
(moyen âge)

Automates

Métiers à tisser
automatiques
Bouchon (1725)
Vaucansson (1748)
Jacquard (1810)

Machine analytique
Babbage (1840)

© F. Anceau, janvier 2002, Page 38

ANTIQUITE

TECHNOLOGIE

Mécanique

Electromécanique

Electronique
Tubes à vide
de Forest (1906)

Additionneur binaire
Stibitz (1937)

Calculateur binaire
programmable V1
Zuse (1938)

Calculateurs
électroniques
programmables
ABC

Atanasoff (1939)
ENIAC BINAC
Eckert, Mauchy (1947- 49)
Automatic Computing Engine
Turing (1950)

THEORIE

Représentation
binaire
Boole (1847)

Théorie du calcul
Turing (1936)

Programme enregistré
Von Neumann (1945)

TEMPS MODERNES

TECHNOLOGIE

Transistors
Schockley, Brattain,
Bardeen (1947)

Mémoires à tores
Papian (1950)

Circuits intégrés
Killy (1958)

Ordinateurs à
transistors
(~1960)

Mini-ordinateurs
MIT LINC
(1962)

Ordinateurs à
circuits intégrés
(~1965)

Microprocesseurs
Intel 4004
Hoff (1971)

L'ordinateur
reste une
machine à
calculer

L'ordinateur
devient
une machine
à traiter de
l'information