La mémoire

SE – Chapitre 4

I. Caractéristiques

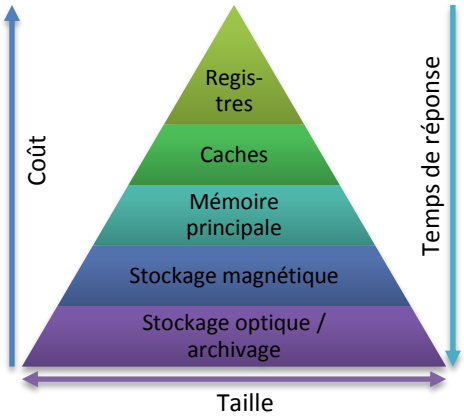
- **Méthode d'accès :**
  séquentiel (bande) / direct (mémoire principale) / mixte (disque) / associatif (cache)
- **Supports :** semi-conducteur / magnétique / optique
- **Volatile** ou non
- **Effaçable** ou non
- **Capacité :** taille du « mot » (svt octet) et nombre de mots
- **Unité de transfert :** mot / bloc / page / fichier / …
- **Performances :** temps d'accès / temps de cycle / debit

II. Exemples

- **RAM :** Random Access Memory (accès direct, volatile)
- **[[[E] E] P]ROM :** [[[Electricaly] Erasable] Programmable] Read Only Memory
  (accès direct, non volatile)

III. Hiérarchie mémoire

- Grande capacité ⇒ lentes
- Rapides ⇒ chères
- Les données passent entre niveaux adjacents



Performances :

- Taux de succès = % d'accès où l'info est au niveau sup.
- Taux d'échec = 1 – taux de succès
- Temps de succès = temps d'accès au niveau sup.
- Pénalité d'échec = temps de remplacement d'un bloc au niveau sup.
- Temps d'accès moyen = temps de succès + taux d'échec × pénalité d'échec

Principe de localité :

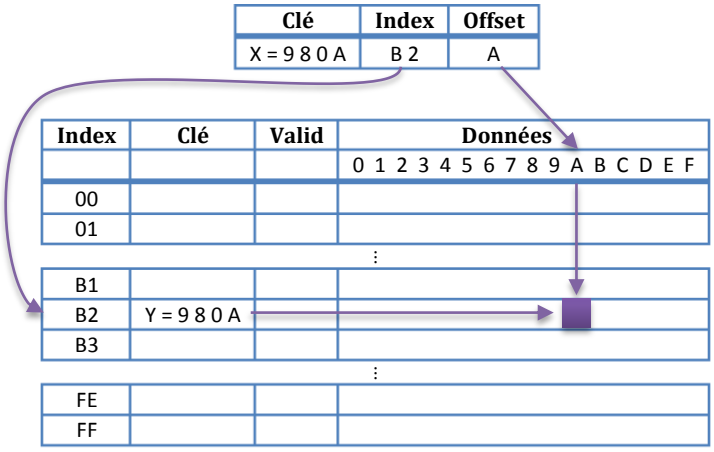
- $\forall\, \Delta t$, un programme accède à une partie relativement petite de son espace d'adressage.
- **Temporelle :** Si un mot est accédé, on a de forte chance d'y accéder à nouveau.
- **Spatiale :** Si un mot est accédé, on a de forte chance d'accéder à ses voisins.

IV. La mémoire cache

1. Structure

a. Cache direct

L'adresse du bloc dans la mémoire principale détermine où il sera rangé dans la mémoire.

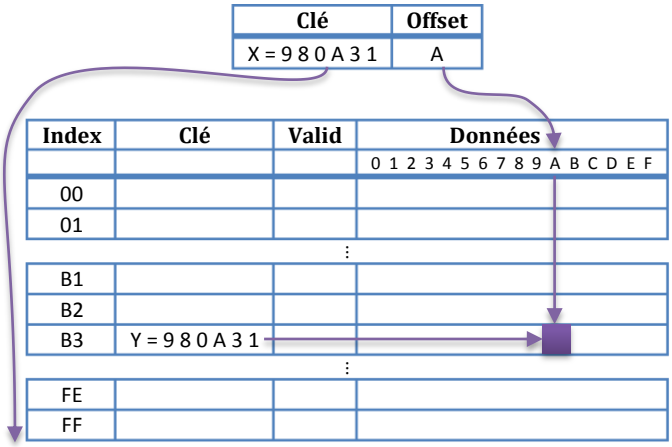


Lecture : si X = Y et valide = 1.
Ecriture : on remplace le bloc.

Avantage : décision rapide
Inconvénient : ping-pong (conflit entre 2 données pour un index cache)

b. Cache associatif

On cherche la clé dans tous les index du cache.



Lecture : si X est dans le cache et valide = 1.
Ecriture : politique de remplacement (Least Recently Used / aléatoire)

Avantage : pas de ping-pong
Inconvénient : décision plus lente

c. Cache associatif à n-voies

Principe du cache direct avec n caches (recherche si la clé est bonne pour chaque cache i).

2. Cohérence cache/mémoire principale

- **Ecriture en cas de succès cache (donnée déjà en cache) :**
  - **Ecriture immédiate :** lent
  - **Ecriture différée :**
    écriture en mémoire quand donnée supprimée du cache (besoin d'un bit « modifié »)
- **Ecriture en cas d'échec cache (donnée pas en cache) :**
  - **Ecriture allouée :** on ajoute le bloc dans le cache
  - **Ecriture non-allouée :** on écrit en mémoire, on ne met rien en cache