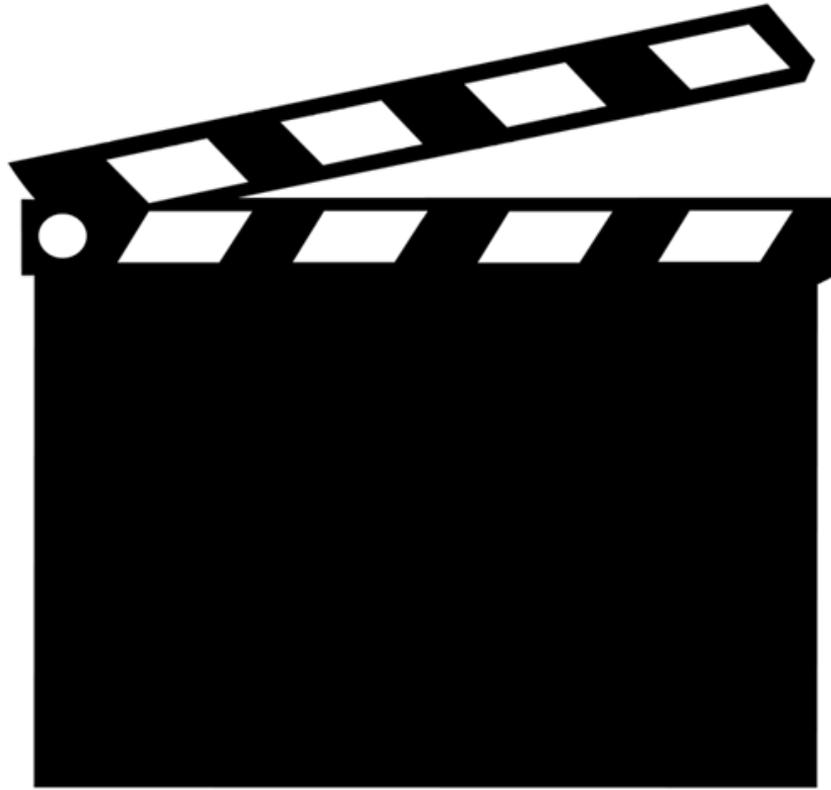


C018SA-W2-S6



SEMAINE 2 : Indexation

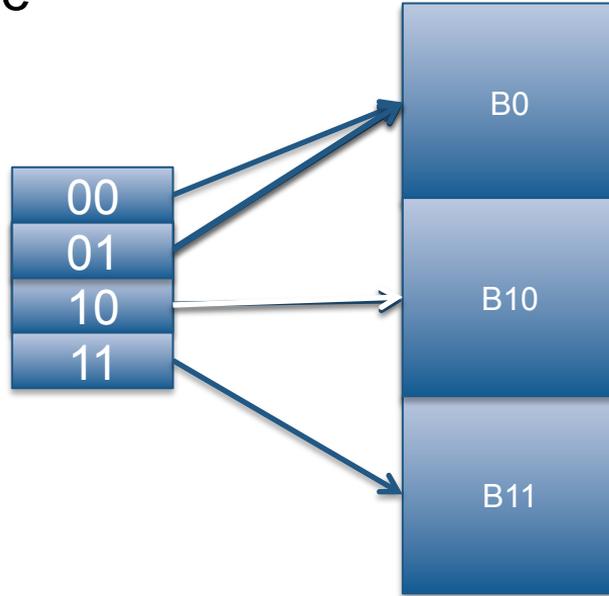
Introduction

2. Hiérarchie de mémoire
3. Fichiers indexés
4. Arbre-B
5. Hachage
- 6. Hachage dynamique**
7. Multi-hachage

Hachage dynamique

- Fonction de hachage retourne n bits
 - $H(k) = b_0b_1b_2\dots b_{n-1}$
- Table de hachage de taille variable : 1,2,4,8...
- Quand un bloc déborde on éclate le bloc en utilisant le bit de hachage suivant

On démarre avec un seul bloc
À une entrée de la table correspond
un bloc unique

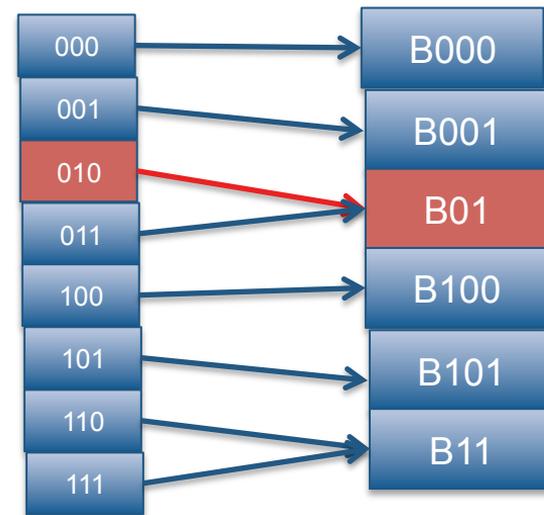


B1 éclate
en B10, B11

B éclate
en B0, B1

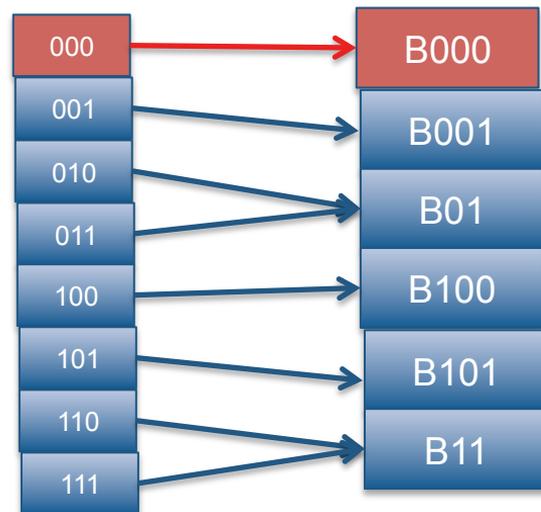
Hachage dynamique

- Après quelques étapes
- Si j'insère dans 010
 - Si j'ai de la place dans B01, juste insertion
 - Sinon, je dois couper le bloc B01
 - ✓ En B010 et B011
 - Je n'ai pas à changer la taille de la table de H



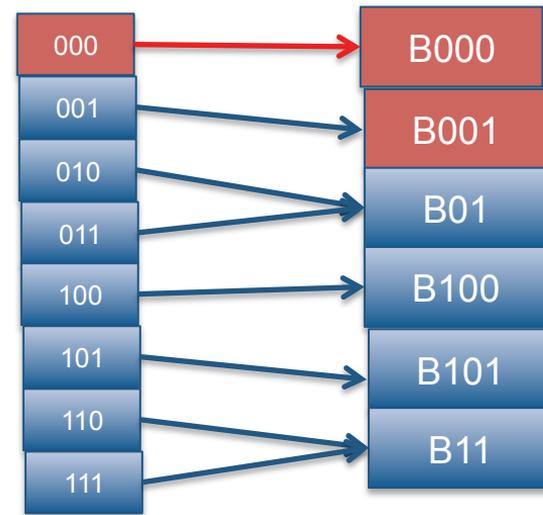
Hachage dynamique

- Après quelques étapes
- Si j'insère dans 000
 - Si j'ai de la place dans B000, juste insertion
 - Sinon je dois doubler la taille de la table



Hachage dynamique

- Après quelques étapes
- Si je supprime dans 000
 - Si B000 et B001 ne sont pas à moitié plein
 - Je les fusionne ?



Hachage dynamique

- Bonne performance
- Travail pour doubler la taille de la table
 - OK car sans lecture disque
- Amortir ce travail : **hachage linéaire**

MERCI