Remplacement de pages

Thomas Lavergne
lavergne@lisn.fr

Coût des défauts de pages

- p = probabilité de défaut de page
- M = temps d'accès à la mémoire
- D = temps de traitement du défaut

$$\text{Temps d'accès} = (1-p) \times M + p \times D = M + p(D-M)$$

Coût des défauts de pages

- p = probabilité de défaut de page
- M = temps d'accès à la mémoire
- D = temps de traitement du défaut

$$\text{Temps d'accès} = (1-p) \times M + p \times D = M + p(D-M)$$

Propriété

Le temps d'accès à la mémoire est, en moyenne, proportionnel à la probabilité de défaut de page.

Problème

Combien de cadres (RAM) allouer à chaque processus

Allocation

Problème

Combien de cadres (RAM) allouer à chaque processus

Propriétés

- Plus de cadre $\longrightarrow$ moins de processus
  $\longrightarrow$ ralentissement
- Moins de cadre $\longrightarrow$ plus de défauts
  $\longrightarrow$ ralentissement

Allocation

Problème

Combien de cadres (RAM) allouer à chaque processus

Propriétés

- Plus de cadre $\rightarrow$ moins de processus
  $\rightarrow$ ralentissement
- Moins de cadre $\rightarrow$ plus de défauts
  $\rightarrow$ ralentissement

Politiques d'allocation

- Allocation équitable
- Allocation proportionnelle
- Allocation basée sur la priorité

Allocation équitable

Données

- N cadres disponibles en RAM
- P processus

Allocation équitable

Données

- N cadres disponibles en RAM
- P processus

Allocation équitable

- Chaque processus reçoit N/P cadres
- Le reste sert de tampon pour supprimer de manière différée

Allocation équitable

Données

- N cadres disponibles en RAM
- P processus

Allocation équitable

- Chaque processus reçoit N/P cadres
- Le reste sert de tampon pour supprimer de manière différée

Inconvénient

Tous les processus n'ont pas besoin de la même quantité de mémoire...

Allocation proportionnelle

Données

- N cadres disponibles en RAM
- P processus
- $\forall i \in [1, P]$, M_i la taille de P_i (en mémoire virtuelle)

Allocation proportionnelle

Données

- N cadres disponibles en RAM
- P processus
- $\forall i \in [1, P]$, M_i la taille de P_i (en mémoire virtuelle)

Allocation proportionnelle

- On garde généralement un tampon de T pages
- P_i reçoit $(N - T) \times M_i / \sum_i M_i$ cadres

Allocation proportionnelle

Données

- N cadres disponibles en RAM
- P processus
- $\forall i \in [1, P]$, M_i la taille de P_i (en mémoire virtuelle)

Allocation proportionnelle

- On garde généralement un tampon de T pages
- P_i reçoit $(N - T) \times M_i / \sum_i M_i$ cadres

Inconvénient

Les petits processus font plus de défauts de pages en pratique... *car ils utilisent simultanément une plus grande part de leur code que les gros processus.*

Allocation basée sur la priorité

Données

- N cadres disponibles en RAM
- P processus
- $\forall i \in [1, P]$, $prio_i$ la priorité de P_i

Allocation basée sur la priorité

Données

- N cadres disponibles en RAM
- P processus
- $\forall i \in [1, P]$, $prio_i$ la priorité de P_i

Allocation basée sur la priorité

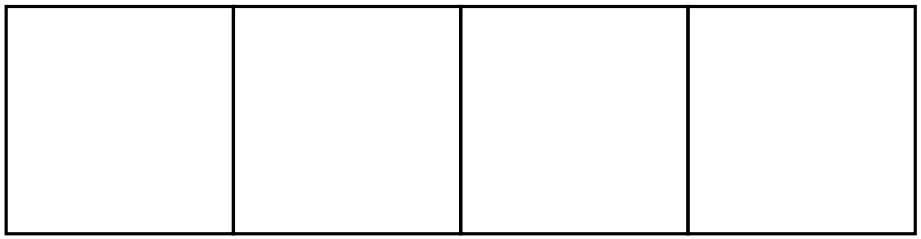
- On garde généralement un tampon de T pages
- P_i reçoit $(N - T) \times prio_i / \sum_i prio_i$ cadres

Mémoire virtuelle

Chaque processus dispose d'un nombre de cadres limité

- Libérer un cadre lorsqu'on a besoin d'une nouvelle page

Exemple:



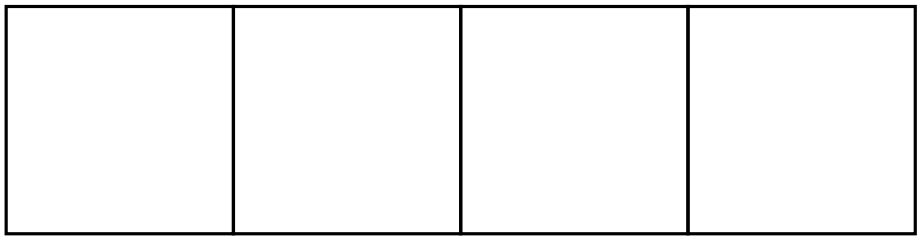
Défauts:

Mémoire virtuelle

Chaque processus dispose d'un nombre de cadres limité

- Libérer un cadre lorsqu'on a besoin d'une nouvelle page

Exemple: 03 2A 1F 04



Défauts: 0

Mémoire virtuelle

Chaque processus dispose d'un nombre de cadres limité

- Libérer un cadre lorsqu'on a besoin d'une nouvelle page

Exemple: 03 2A 1F 04

03	2A	1F	04

Défauts: 4

Mémoire virtuelle

Chaque processus dispose d'un nombre de cadres limité

- Libérer un cadre lorsqu'on a besoin d'une nouvelle page

Exemple: 03 2A 1F 04 2A

03	2A	1F	04

Défauts: 4

Mémoire virtuelle

Chaque processus dispose d'un nombre de cadres limité

- Libérer un cadre lorsqu'on a besoin d'une nouvelle page

Exemple: 03 2A 1F 04 2A

03	2A	1F	04

Défauts: 4

Mémoire virtuelle

Chaque processus dispose d'un nombre de cadres limité

- Libérer un cadre lorsqu'on a besoin d'une nouvelle page

Exemple: 03 2A 1F 04 2A 12

03	2A	1F	04

Défauts: 4

Mémoire virtuelle

Chaque processus dispose d'un nombre de cadres limité

- Libérer un cadre lorsqu'on a besoin d'une nouvelle page

Exemple: 03 2A 1F 04 2A 12

12	2A	1F	04

Défauts: 5

Mémoire virtuelle

Chaque processus dispose d'un nombre de cadres limité

- Libérer un cadre lorsqu'on a besoin d'une nouvelle page

Exemple: 03 2A 1F 04 2A 12 03

12	2A	1F	04

Défauts: 5

Mémoire virtuelle

Chaque processus dispose d'un nombre de cadres limité

- Libérer un cadre lorsqu'on a besoin d'une nouvelle page

Exemple: 03 2A 1F 04 2A 12 03

03	2A	1F	04

Défauts: 6

Représentation visuelle

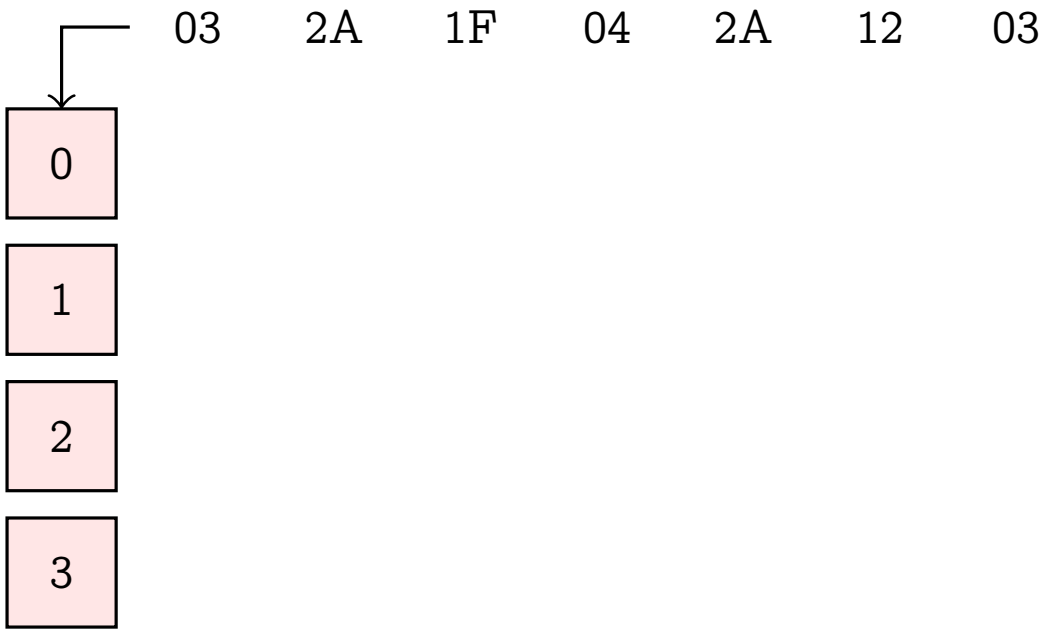
Objectif

Figurer l'état du cache dans le temps pour compter le nombre de défauts

Représentation visuelle

Objectif

Figurer l'état du cache dans le temps pour compter le nombre de défauts

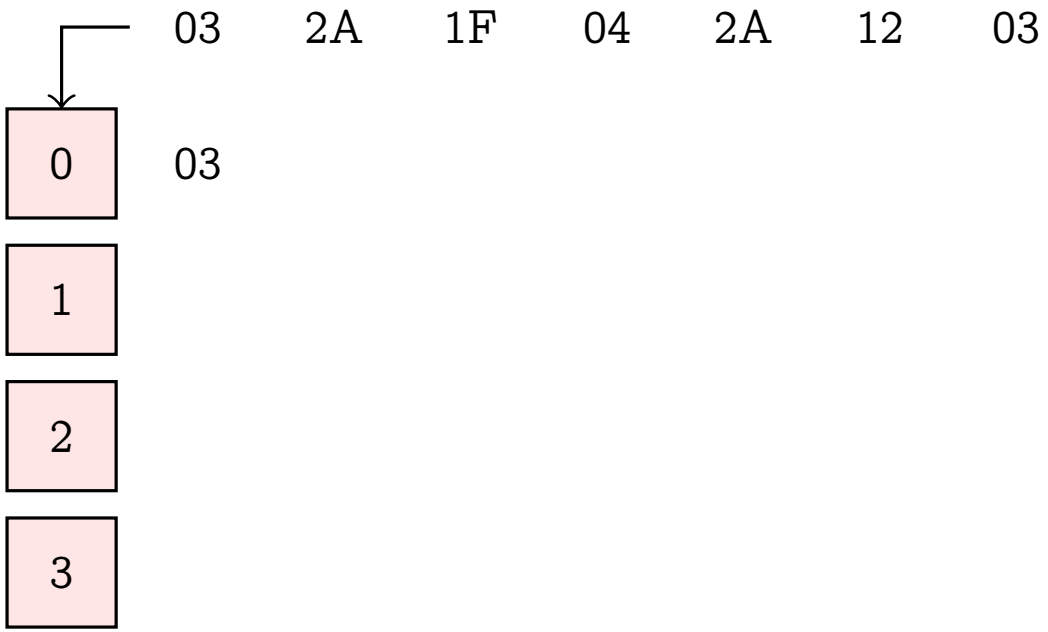


Une ligne par cadre de page alloué au processus

Représentation visuelle

Objectif

Figurer l'état du cache dans le temps pour compter le nombre de défauts

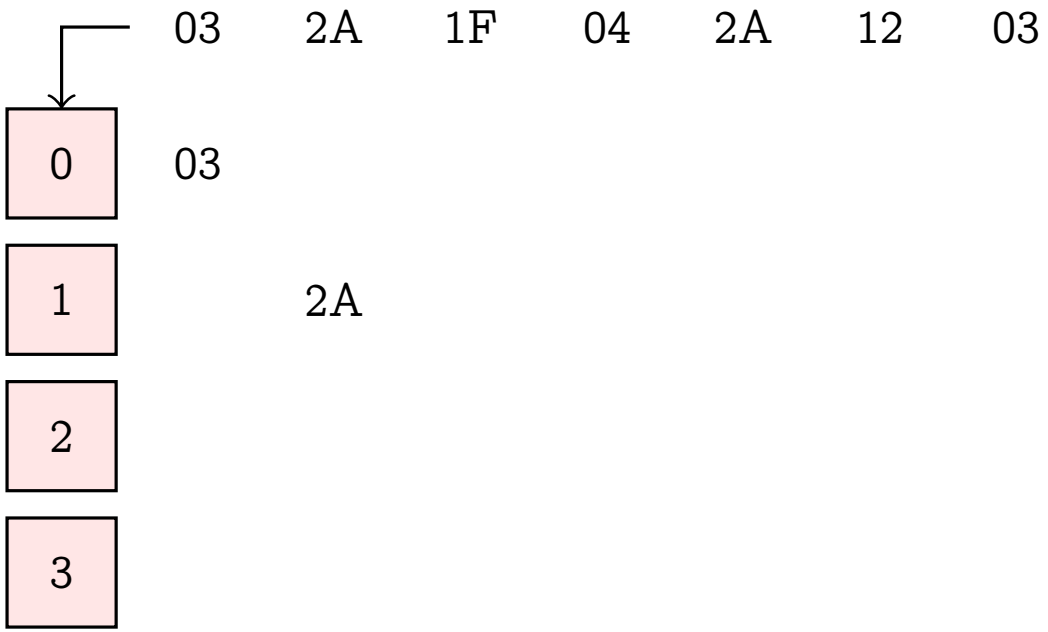


Une page par colonne

Représentation visuelle

Objectif

Figurer l'état du cache dans le temps pour compter le nombre de défauts

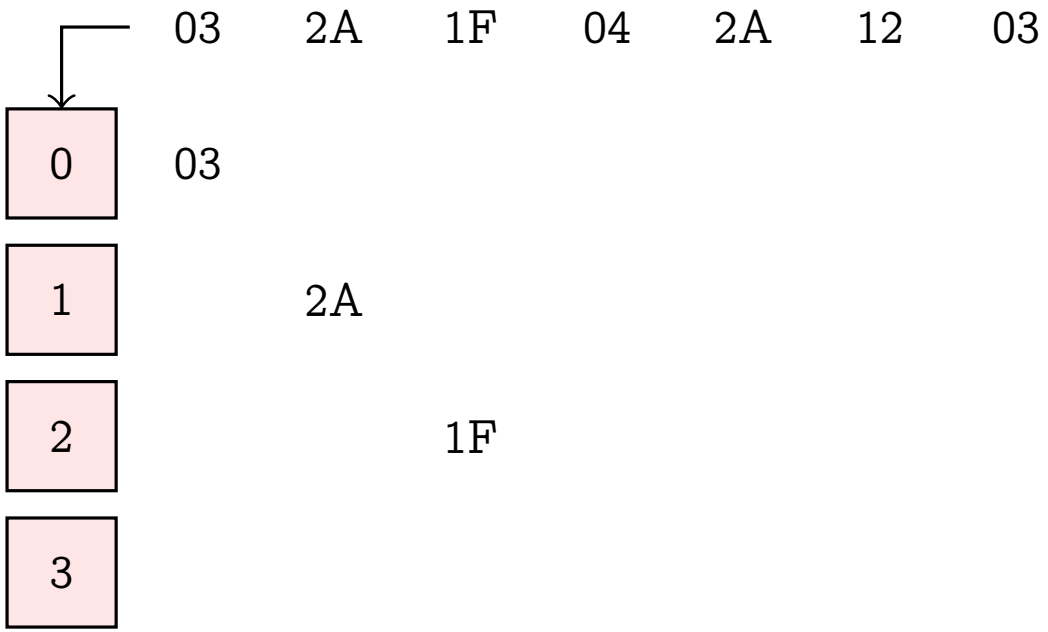


Une page par colonne

Représentation visuelle

Objectif

Figurer l'état du cache dans le temps pour compter le nombre de défauts

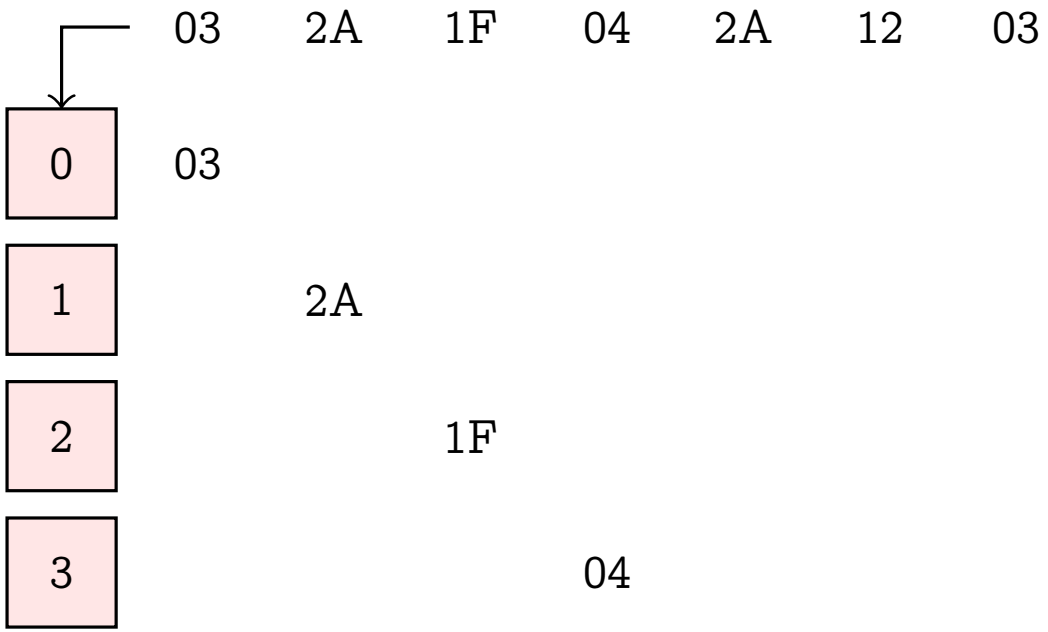


Une page par colonne

Représentation visuelle

Objectif

Figurer l'état du cache dans le temps pour compter le nombre de défauts

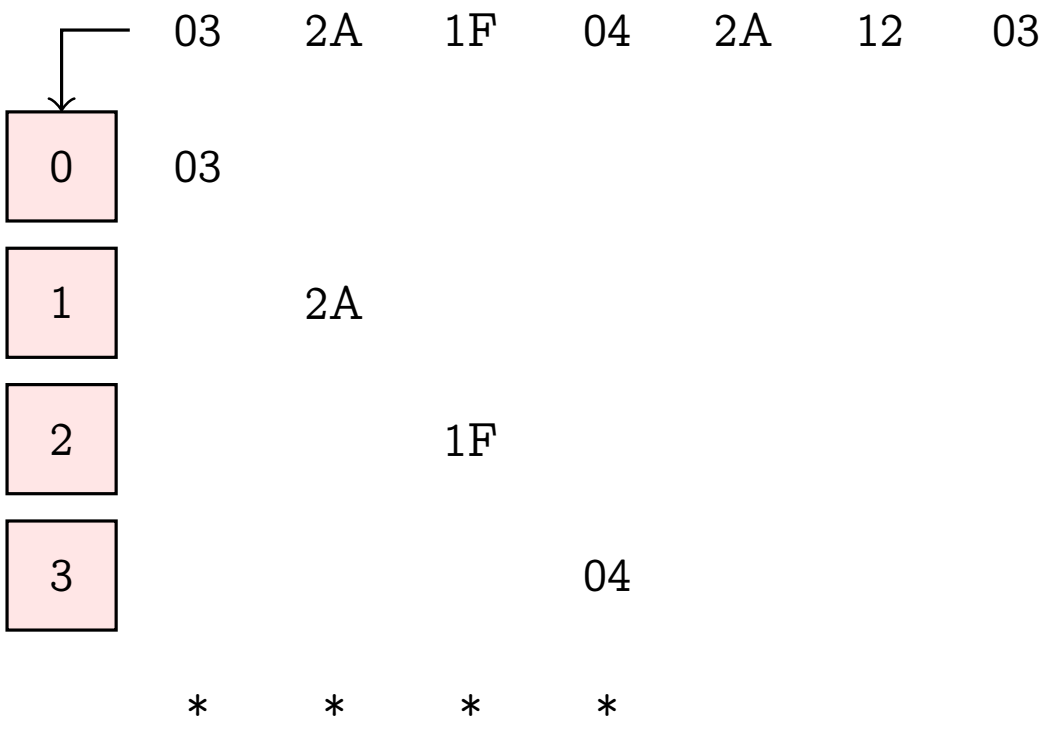


Une page par colonne

Représentation visuelle

Objectif

Figurer l'état du cache dans le temps pour compter le nombre de défauts

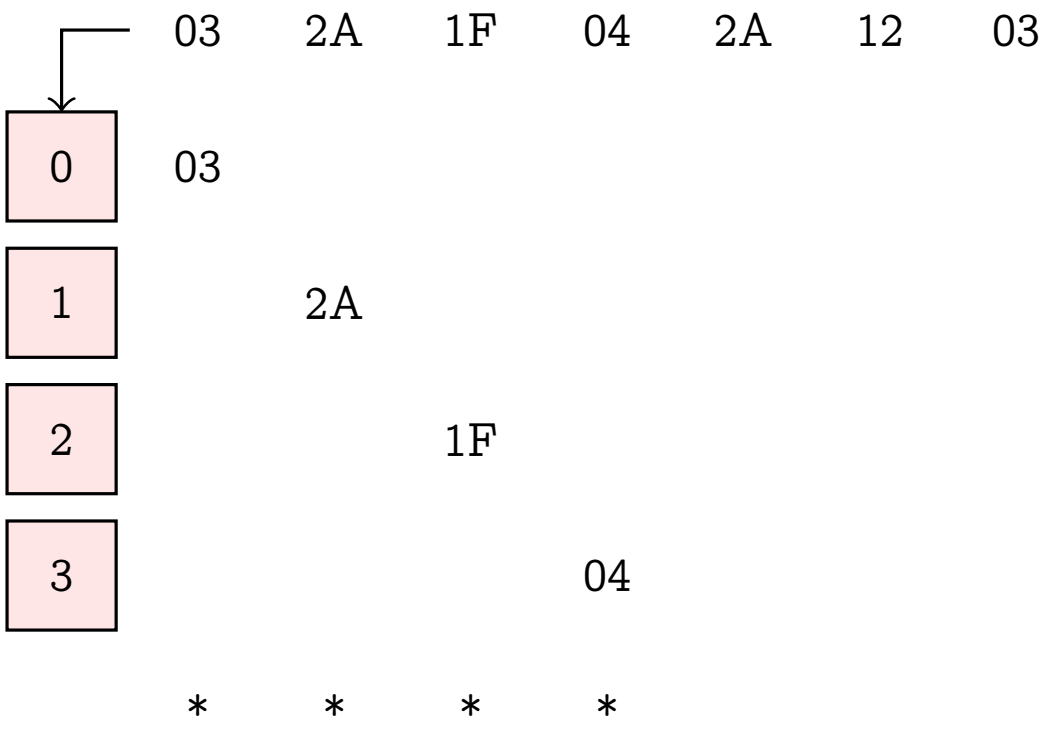


Marquer les défauts au fur et à mesure

Représentation visuelle

Objectif

Figurer l'état du cache dans le temps pour compter le nombre de défauts

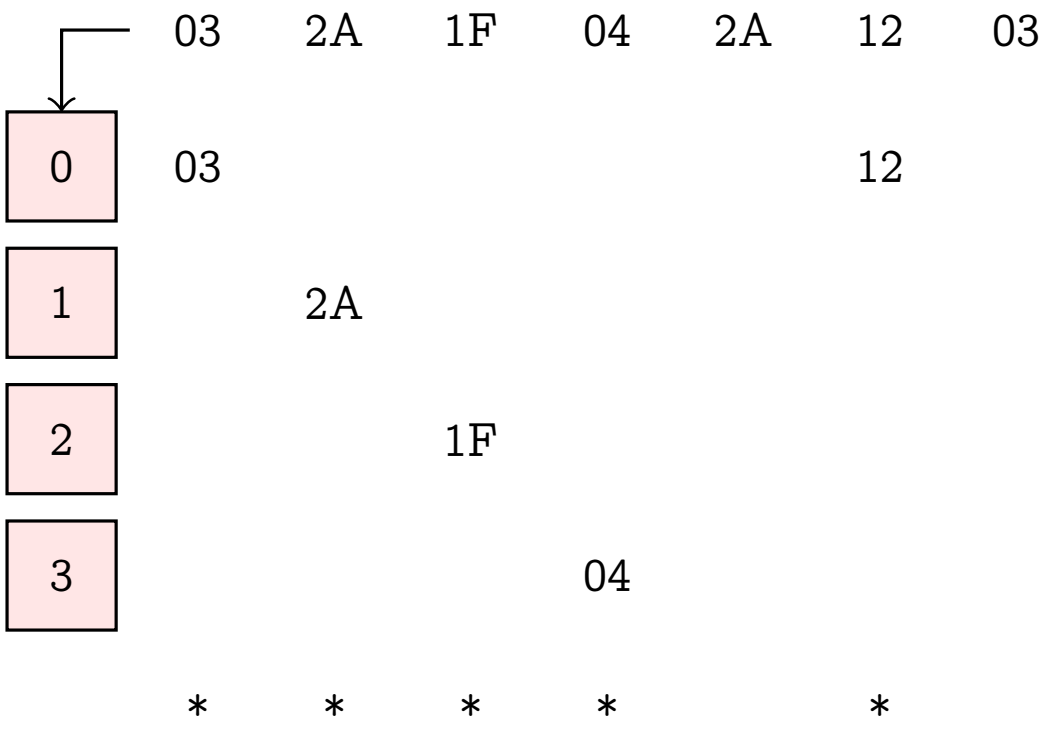


Pas de défaut $\longrightarrow$ *laisser en blanc*

Représentation visuelle

Objectif

Figurer l'état du cache dans le temps pour compter le nombre de défauts

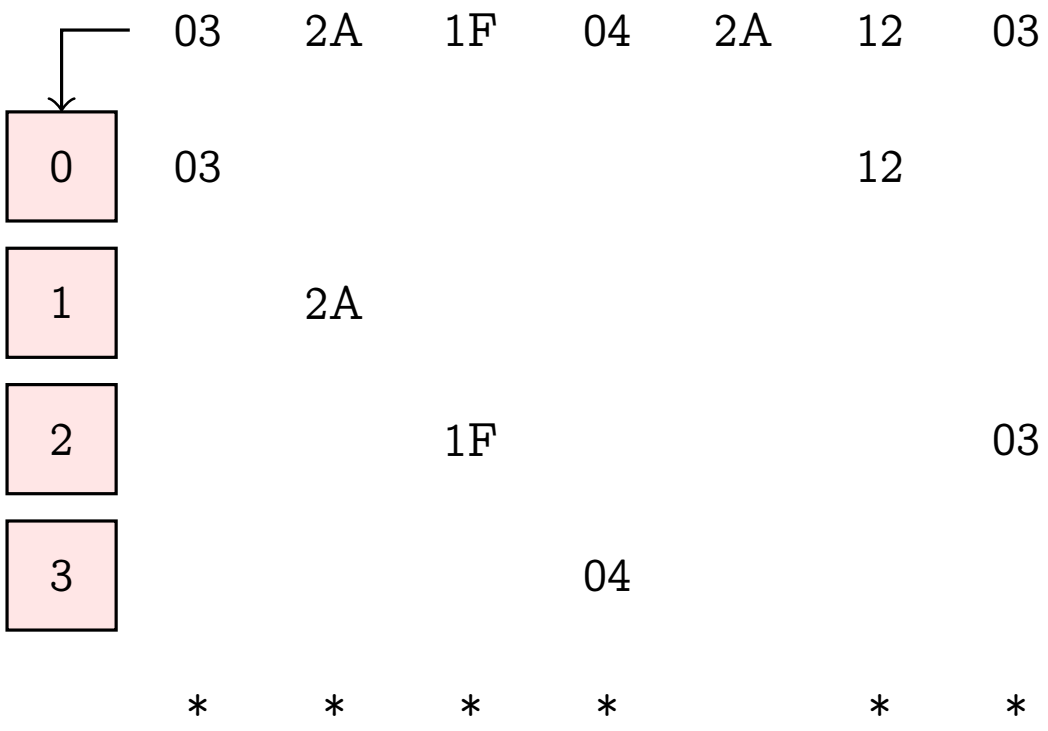


Et on continue...

Représentation visuelle

Objectif

Figurer l'état du cache dans le temps pour compter le nombre de défauts

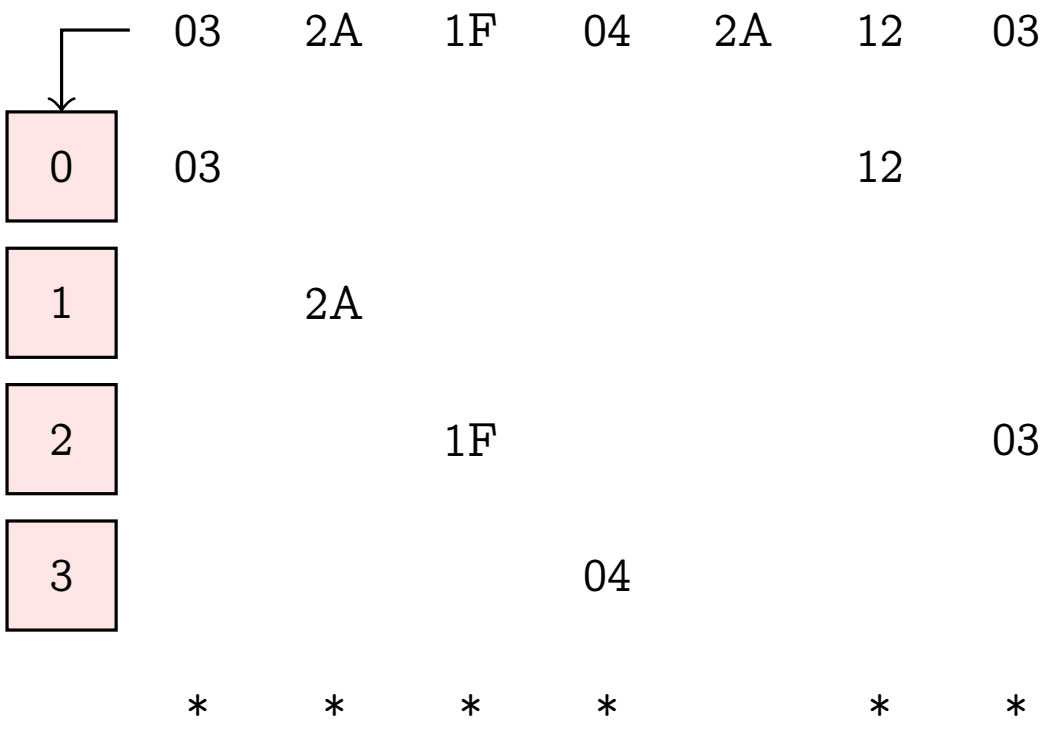


Et on continue...

Représentation visuelle

Objectif

Figurer l'état du cache dans le temps pour compter le nombre de défauts



Total: 6 défauts

Et on continue...

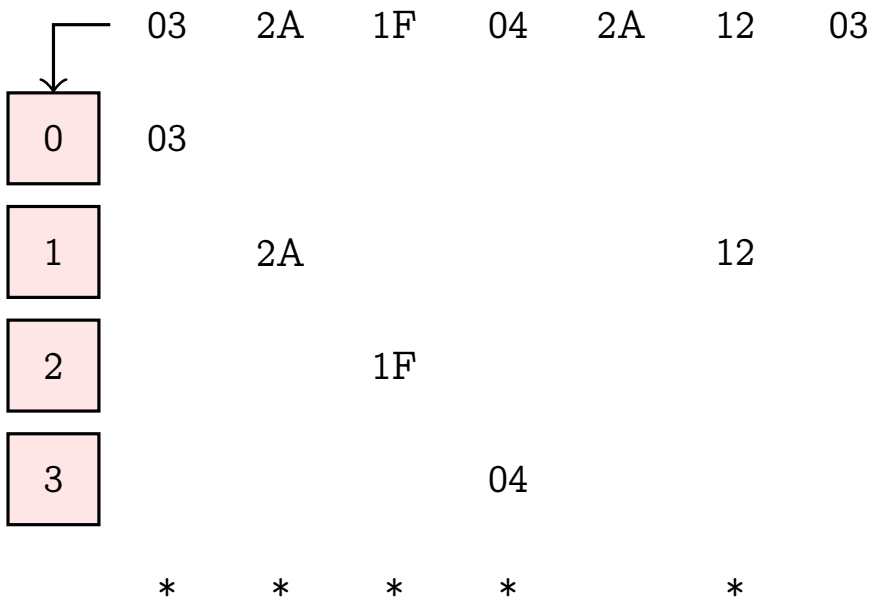
Solution optimale

Politique de remplacement

Pour une instance donnée, il existe une politique de remplacement optimale...

Politique de remplacement

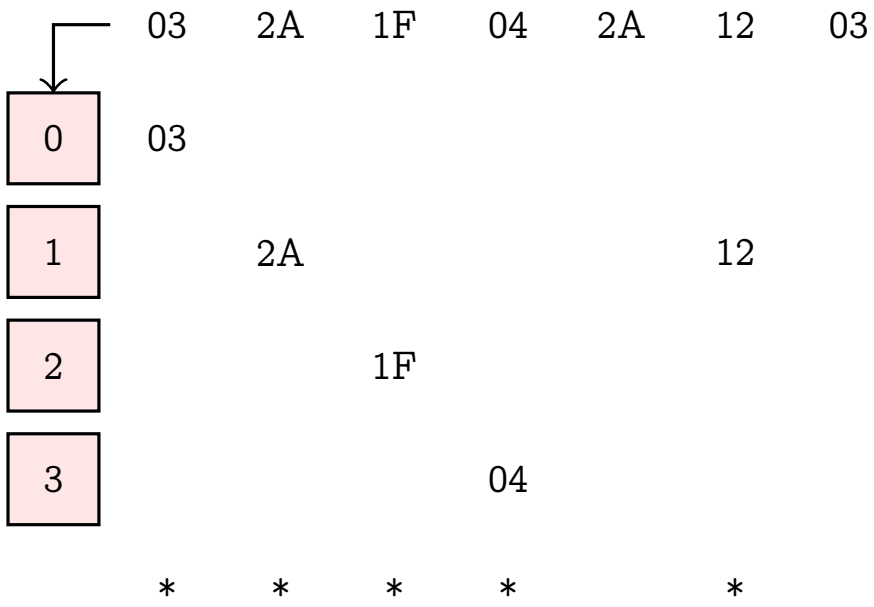
Pour une instance donnée, il existe une politique de remplacement optimale...



Total: 5 *défauts*

Politique de remplacement

Pour une instance donnée, il existe une politique de remplacement optimale... si on connaît à l'avance les pages qui seront demandées!

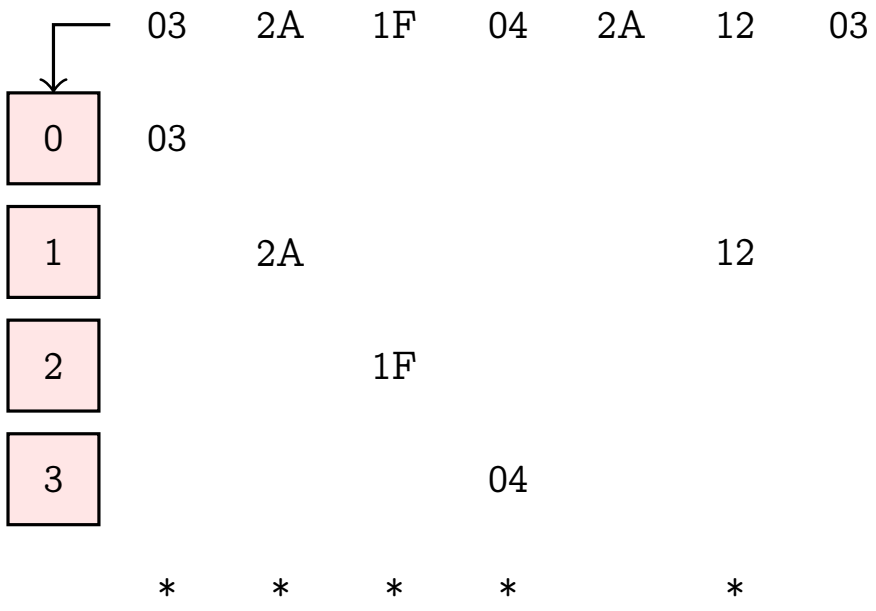


Total: 5 *défauts*

Solution optimale

Politique de remplacement

Pour une instance donnée, il existe une politique de remplacement optimale... si on connaît à l'avance les pages qui seront demandées!



Total: 5 *défauts*

En pratique, on ne connaît pas les pages!

Algorithme de remplacement

Définir une fonction qui, étant donné l'état actuel (occupation des cadres par des pages), décide quel cadre doit être libéré.

Algorithme de remplacement

Définir une fonction qui, étant donné l'état actuel (occupation des cadres par des pages), décide quel cadre doit être libéré.

Classes d'algorithmes

- First In, First Out (FIFO)
  - retirer les pages les plus anciennes
- Basés sur l'utilisation des pages (Least Used)
  - retirer les pages les moins utilisées

Algorithme de remplacement

Définir une fonction qui, étant donné l'état actuel (occupation des cadres par des pages), décide quel cadre doit être libéré.

Classes d'algorithmes

- First In, First Out (FIFO)
  - retirer les pages les plus anciennes
- Basés sur l'utilisation des pages (Least Used)
  - retirer les pages les moins utilisées

Évaluation = nombre de défauts de page

- Sur des instances (en TD, à l'examen)
- Sur des *benchmarks*

Algorithmes FIFO

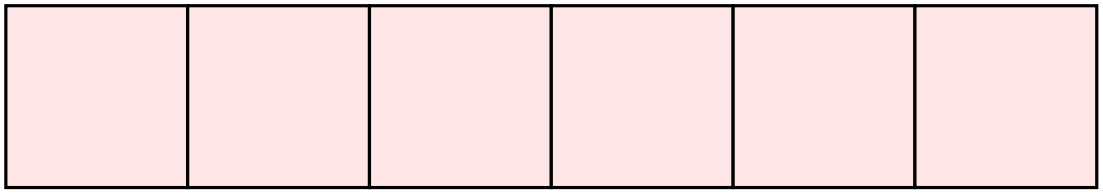
File

Retirer la page la plus ancienne

File

Retirer la page la plus ancienne

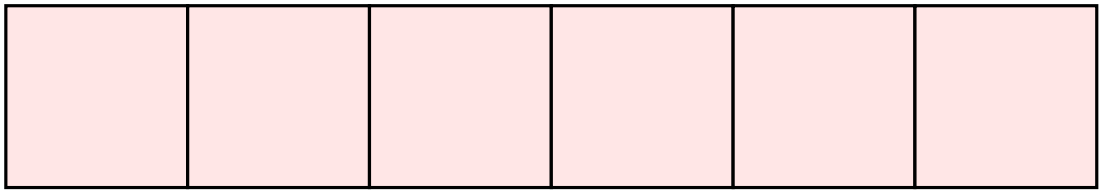
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



File

Retirer la page la plus ancienne

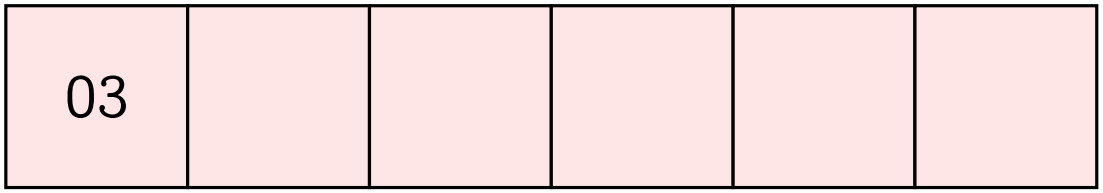
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



File

Retirer la page la plus ancienne

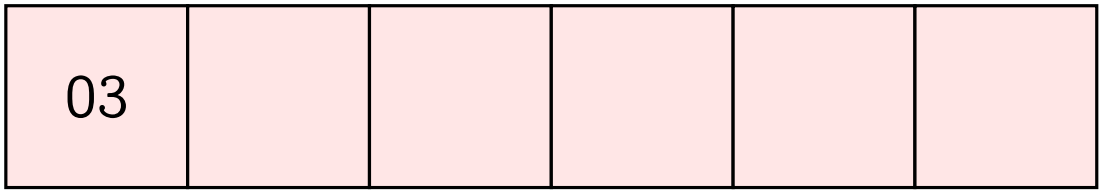
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



File

Retirer la page la plus ancienne

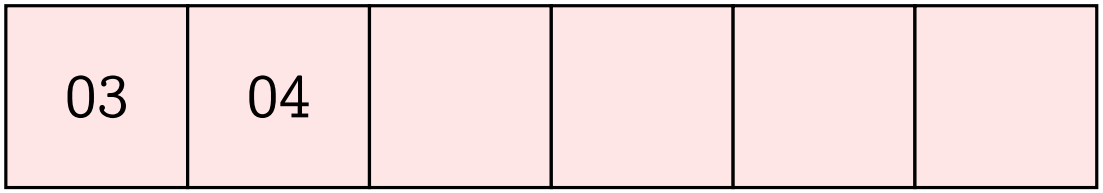
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



File

Retirer la page la plus ancienne

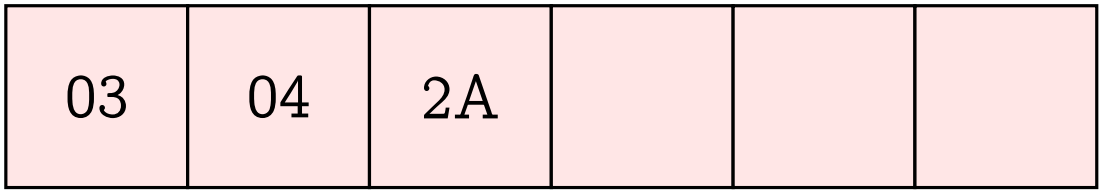
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



File

Retirer la page la plus ancienne

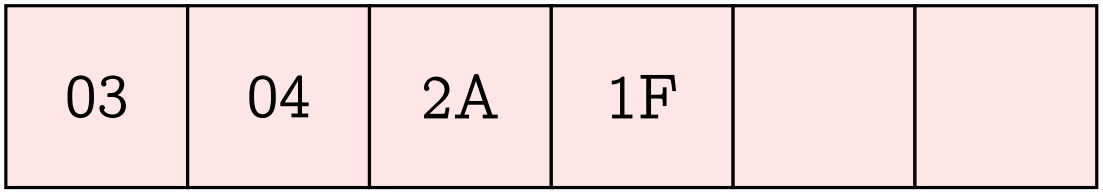
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



File

Retirer la page la plus ancienne

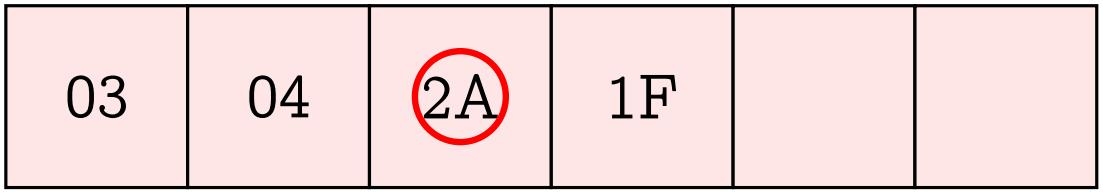
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



File

Retirer la page la plus ancienne

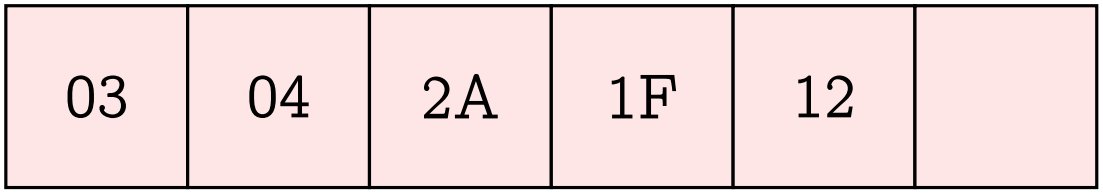
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



File

Retirer la page la plus ancienne

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



File

Retirer la page la plus ancienne

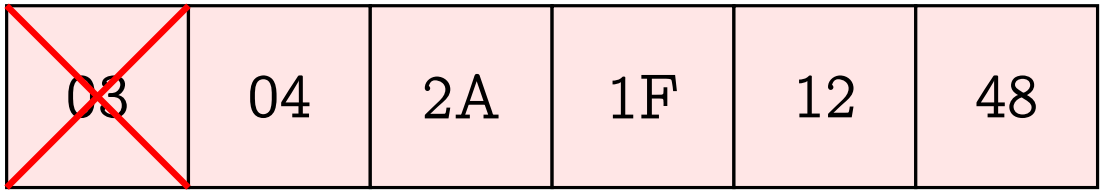
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

03	04	2A	1F	12	48

File

Retirer la page la plus ancienne

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



File

Retirer la page la plus ancienne

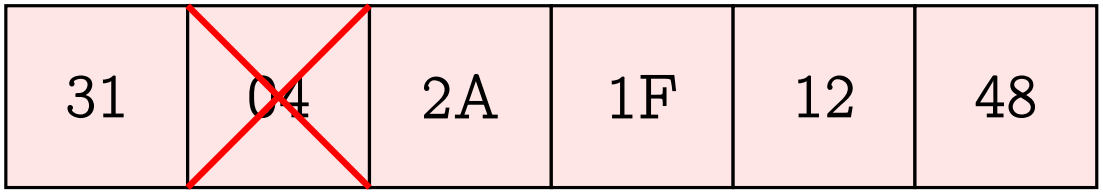
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

31	04	2A	1F	12	48

File

Retirer la page la plus ancienne

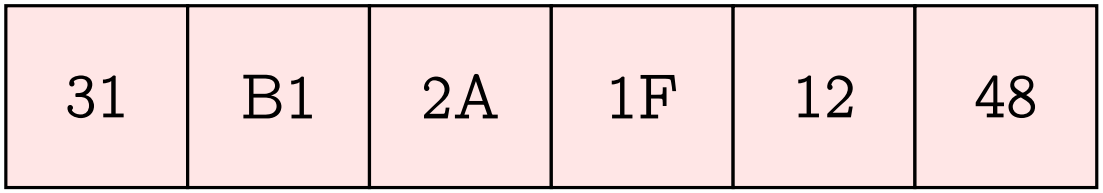
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



File

Retirer la page la plus ancienne

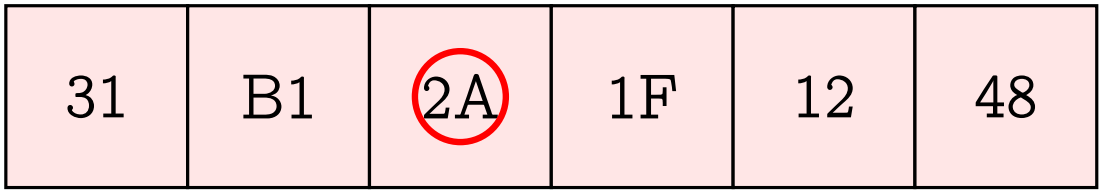
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



File

Retirer la page la plus ancienne

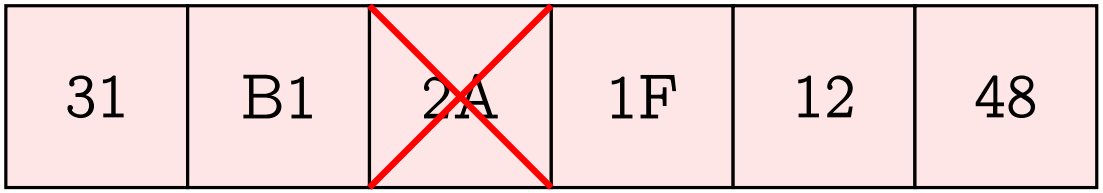
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



File

Retirer la page la plus ancienne

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



File

Retirer la page la plus ancienne

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

31	B1	03	1F	12	48

File

Retirer la page la plus ancienne

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

31	B1	03	76	12	48

File

Retirer la page la plus ancienne

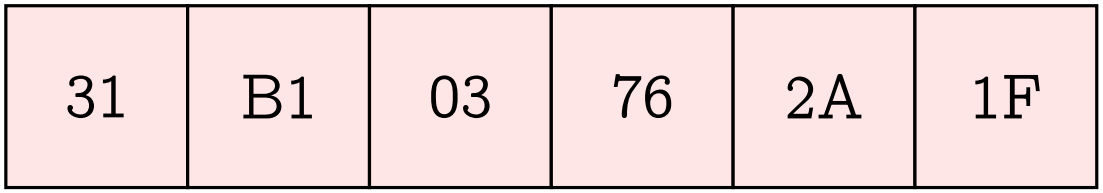
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

31	B1	03	76	2A	48

File

Retirer la page la plus ancienne

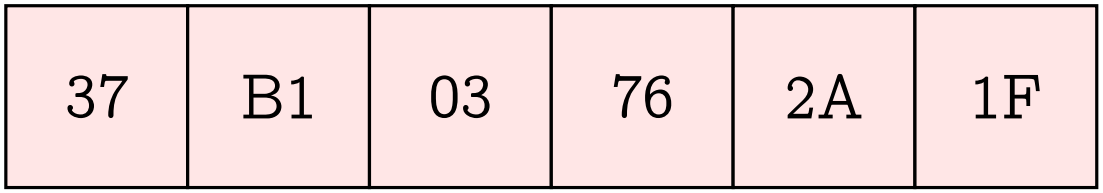
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



File

Retirer la page la plus ancienne

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

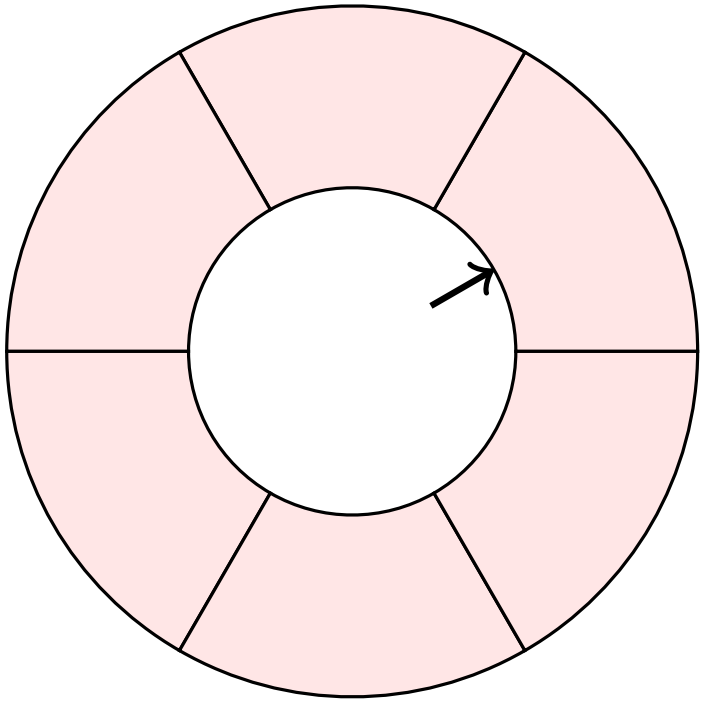


File

Retirer la page la plus ancienne

- Implémentation = liste circulaire (*clock based*)

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

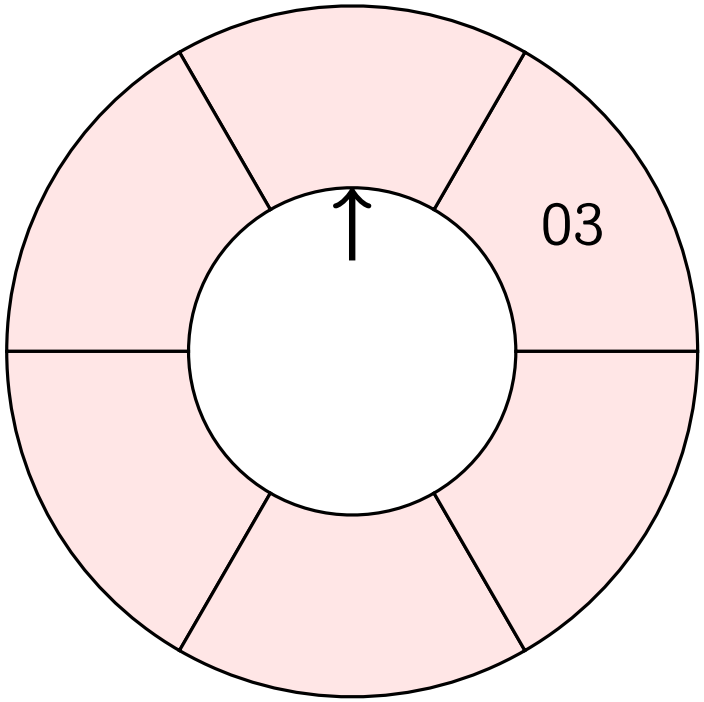


File

Retirer la page la plus ancienne

- Implémentation = liste circulaire (*clock based*)

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

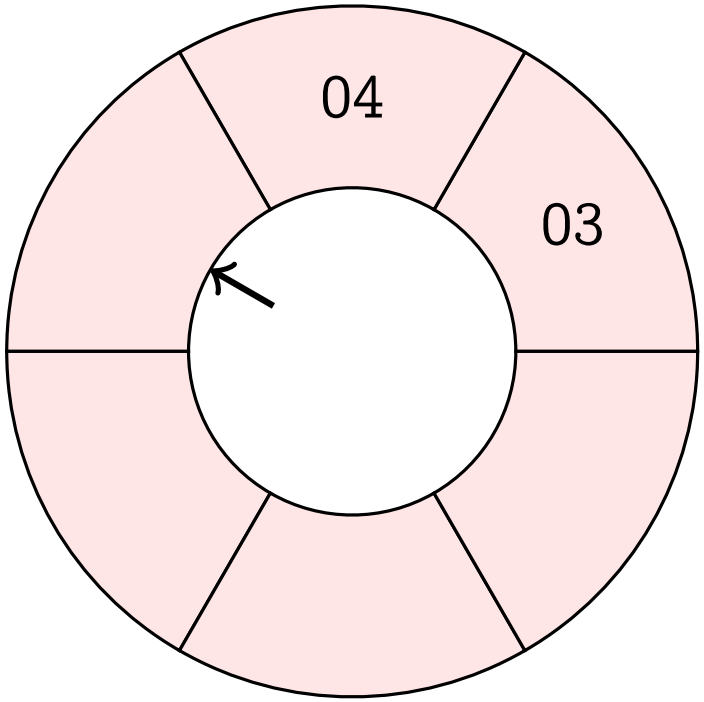


File

Retirer la page la plus ancienne

- Implémentation = liste circulaire (*clock based*)

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

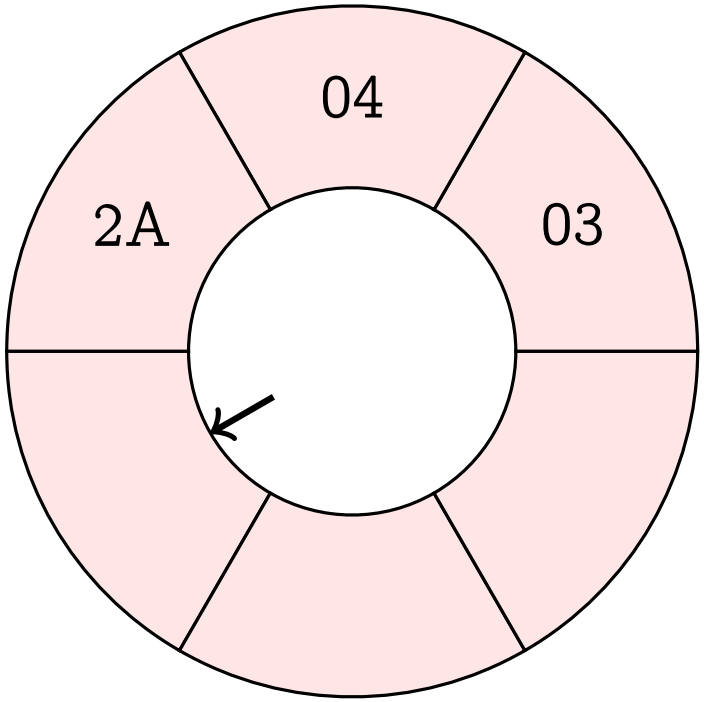


File

Retirer la page la plus ancienne

- Implémentation = liste circulaire (*clock based*)

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

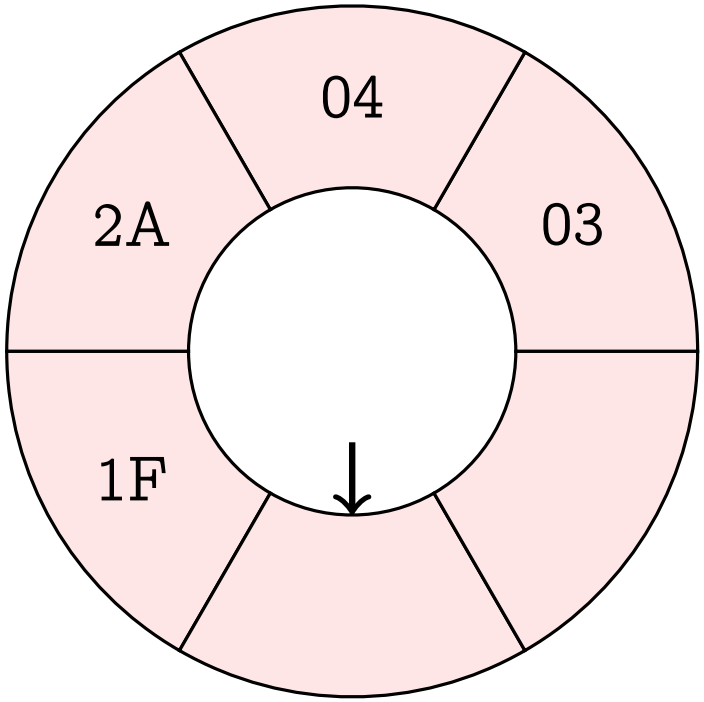


File

Retirer la page la plus ancienne

- Implémentation = liste circulaire (*clock based*)

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

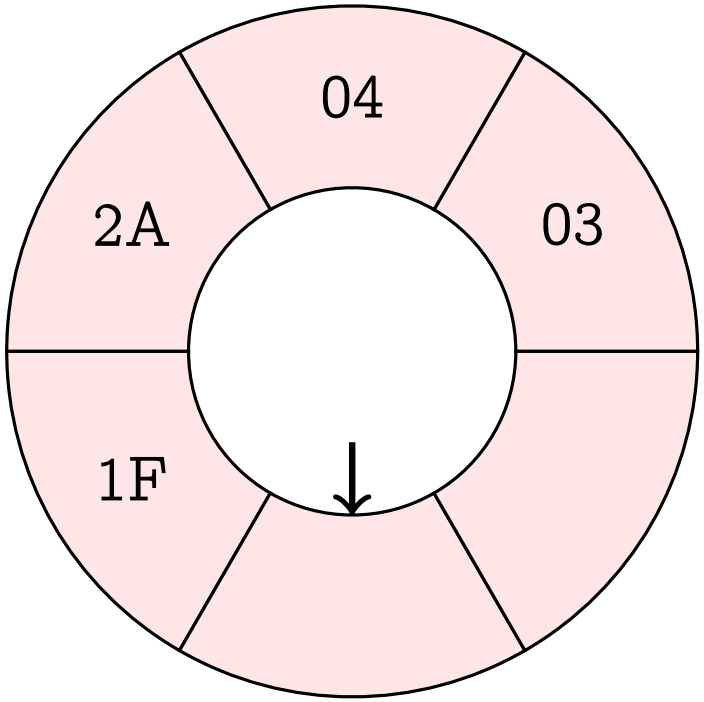


File

Retirer la page la plus ancienne

- Implémentation = liste circulaire (*clock based*)

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

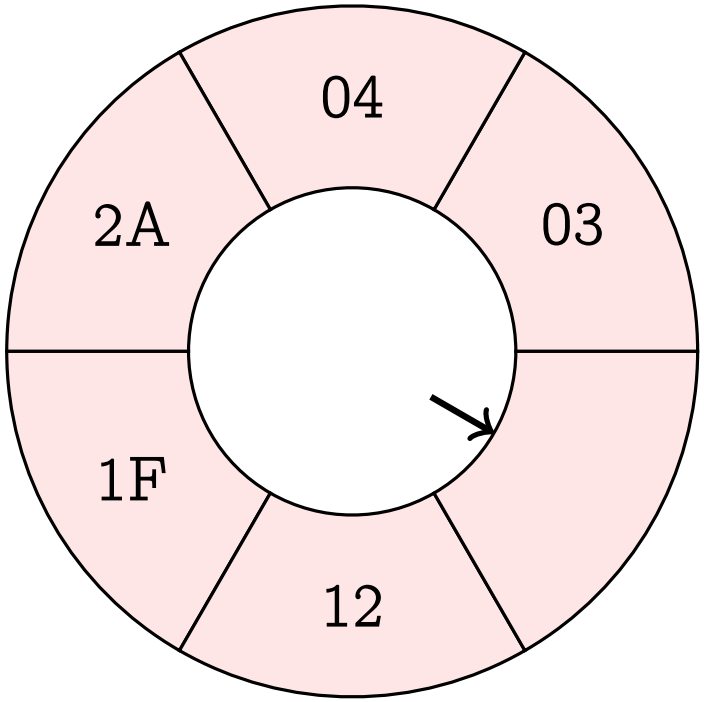


File

Retirer la page la plus ancienne

- Implémentation = liste circulaire (*clock based*)

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

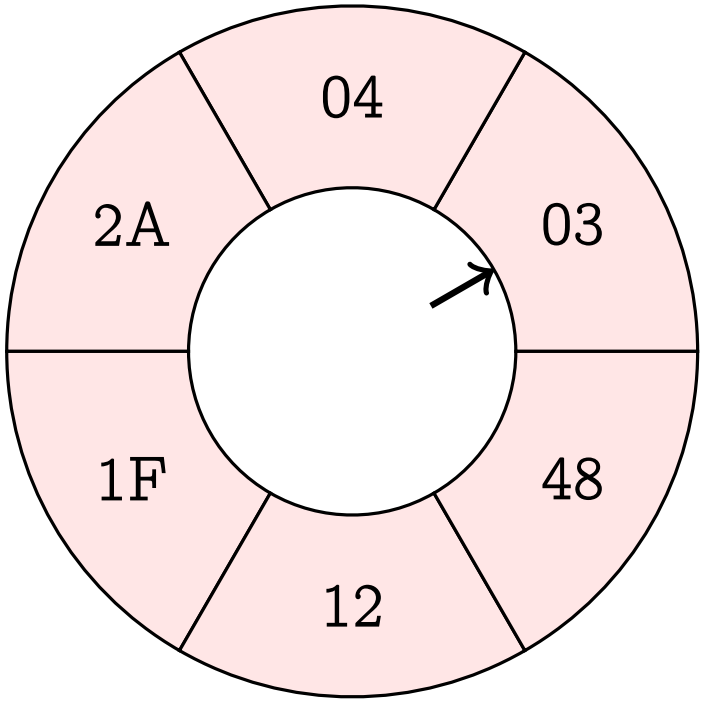


File

Retirer la page la plus ancienne

- Implémentation = liste circulaire (*clock based*)

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

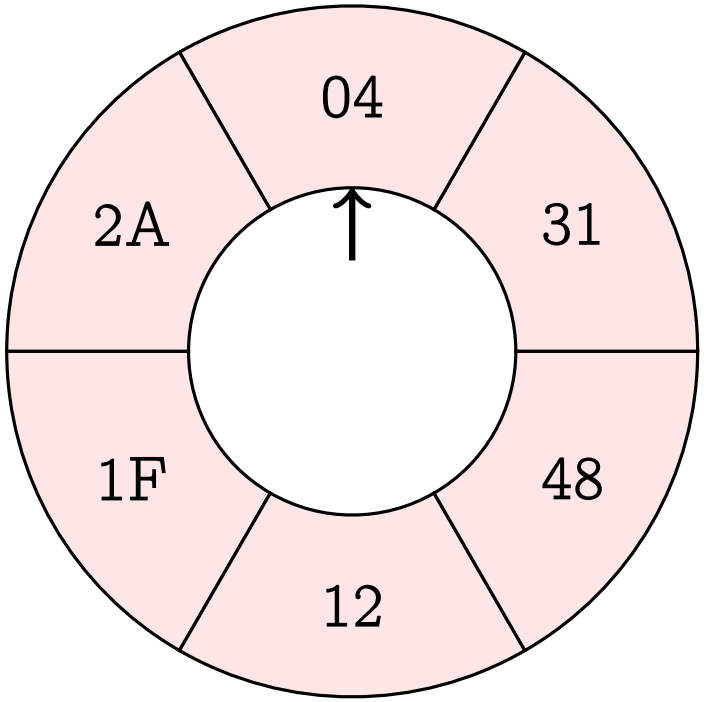


File

Retirer la page la plus ancienne

- Implémentation = liste circulaire (*clock based*)

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

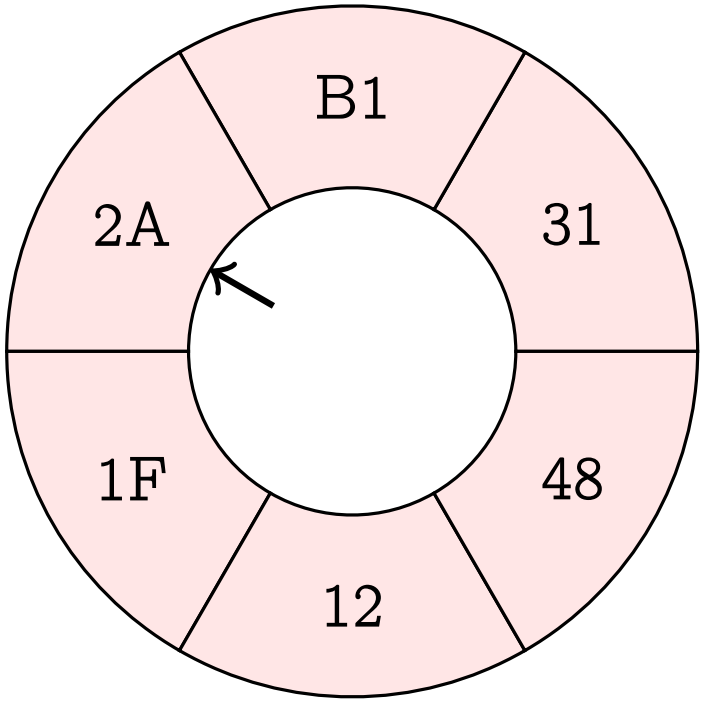


File

Retirer la page la plus ancienne

- Implémentation = liste circulaire (*clock based*)

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

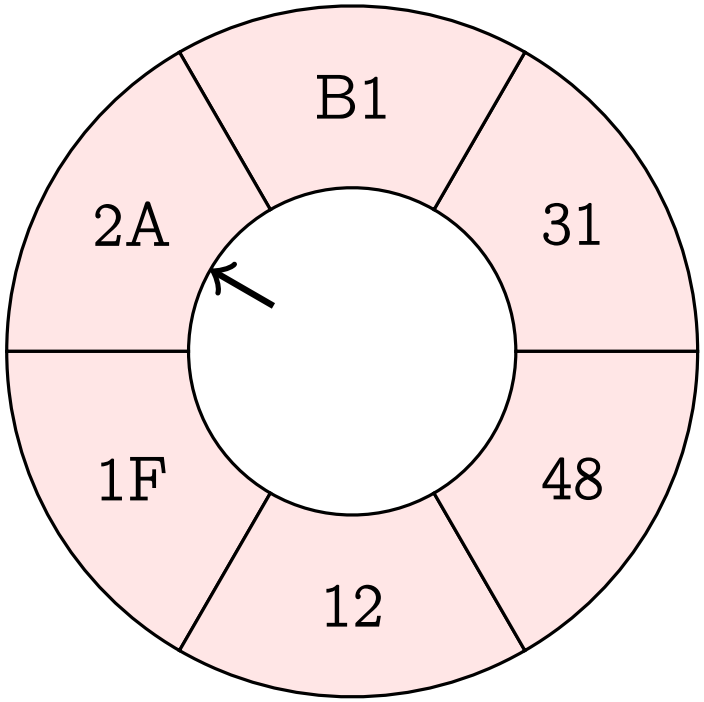


File

Retirer la page la plus ancienne

- Implémentation = liste circulaire (*clock based*)

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

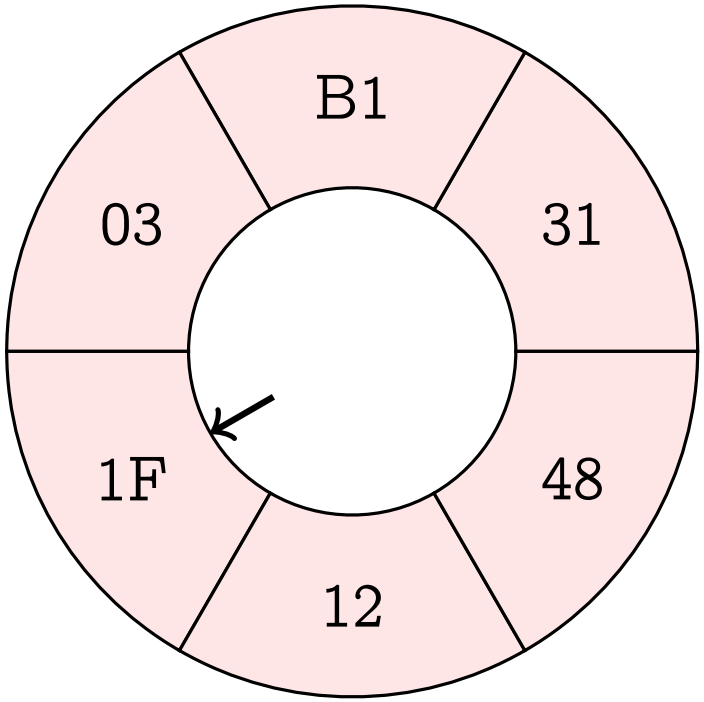


File

Retirer la page la plus ancienne

- Implémentation = liste circulaire (*clock based*)

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

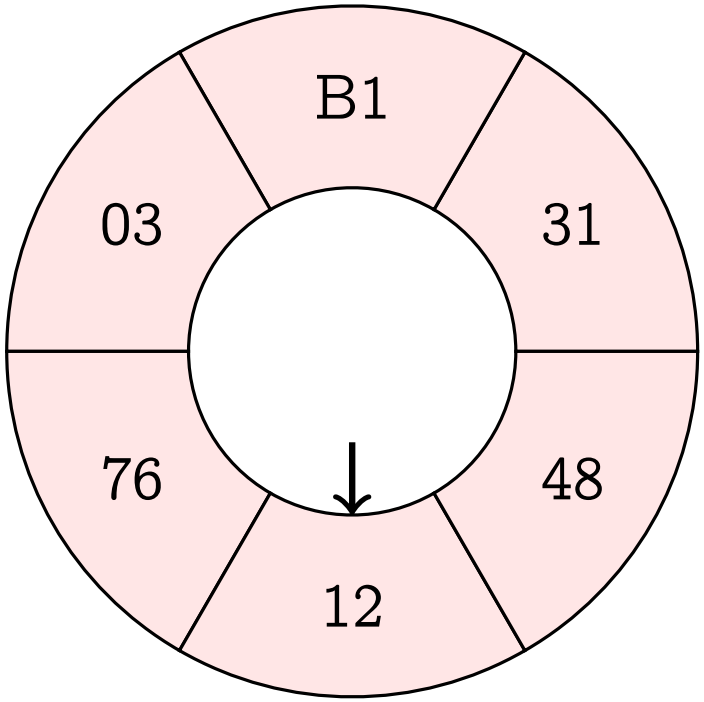


File

Retirer la page la plus ancienne

- Implémentation = liste circulaire (*clock based*)

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

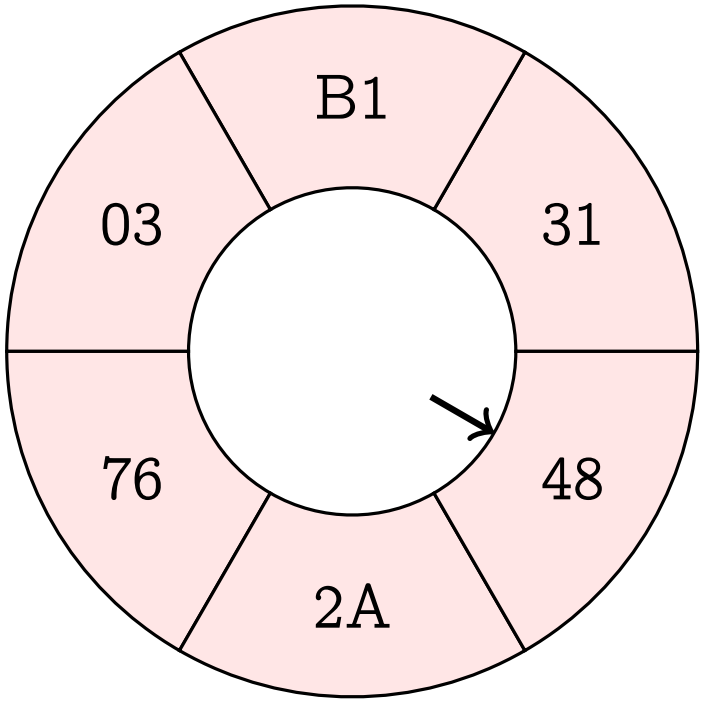


File

Retirer la page la plus ancienne

- Implémentation = liste circulaire (*clock based*)

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

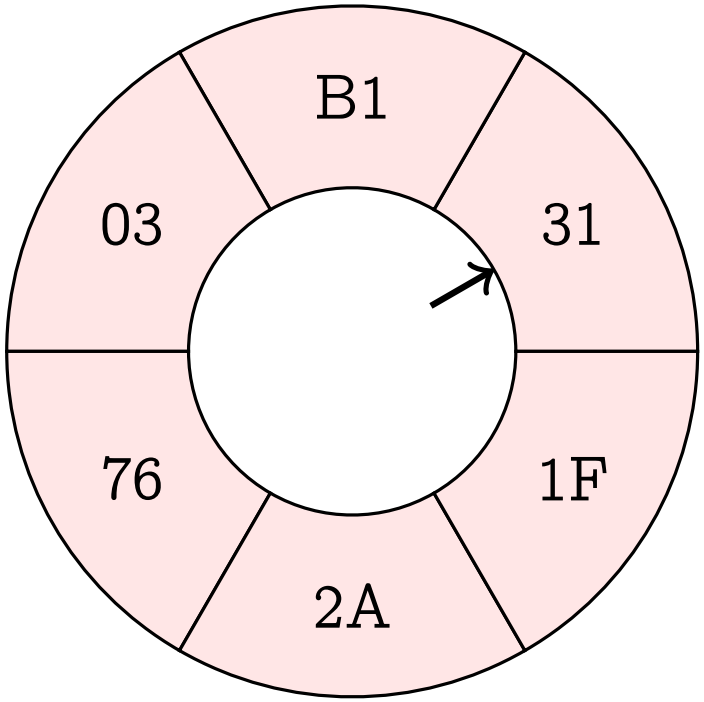


File

Retirer la page la plus ancienne

- Implémentation = liste circulaire (*clock based*)

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

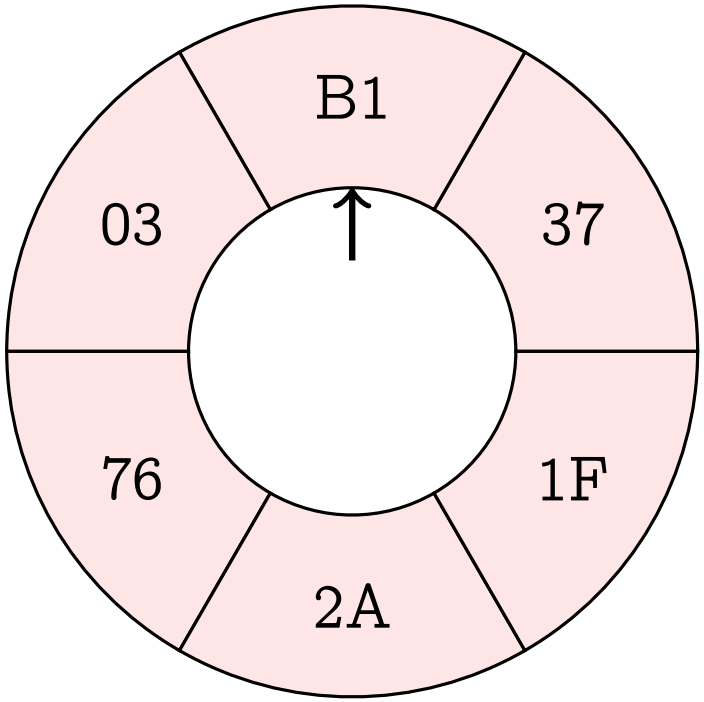


File

Retirer la page la plus ancienne

- Implémentation = liste circulaire (*clock based*)

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

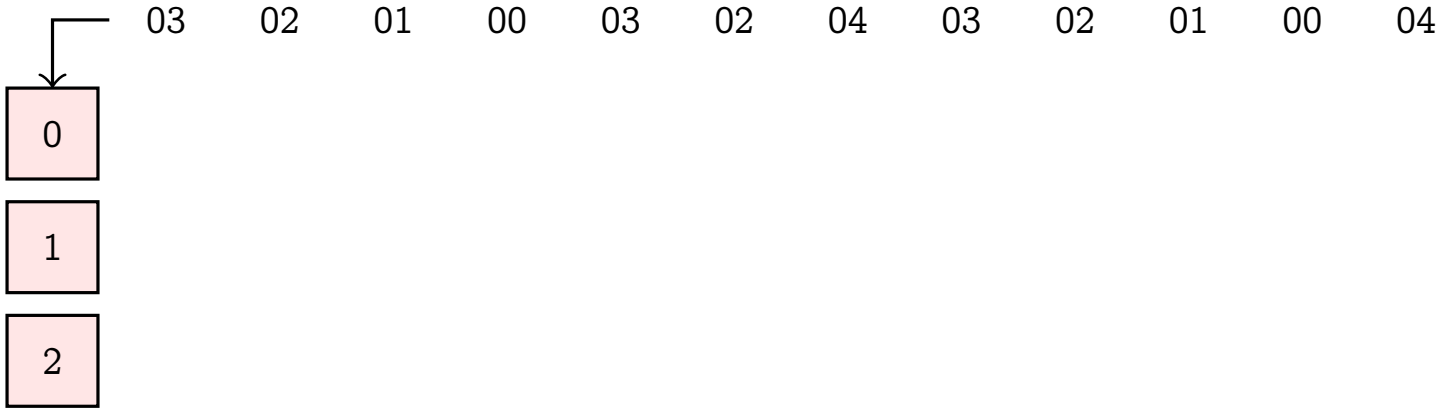


Algorithmes de type FIFO

Sur certaines instances...

Plus de cadres $\rightarrow$ plus de défauts!

Exemple: avec 3 cadres

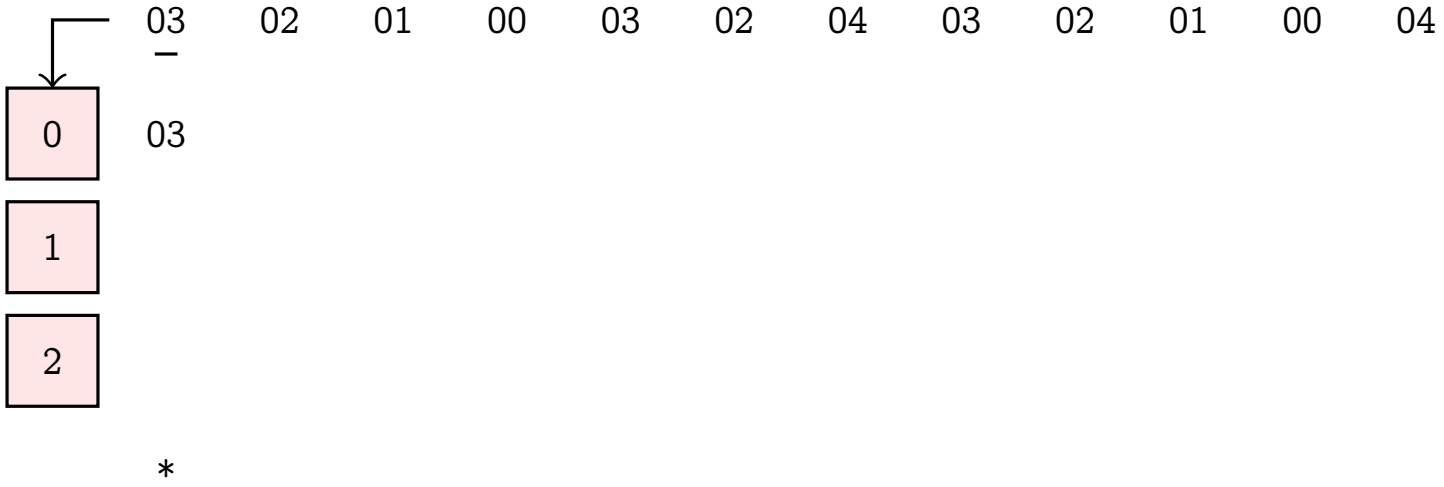


Algorithmes de type FIFO

Sur certaines instances...

Plus de cadres $\longrightarrow$ plus de défauts!

Exemple: avec 3 cadres

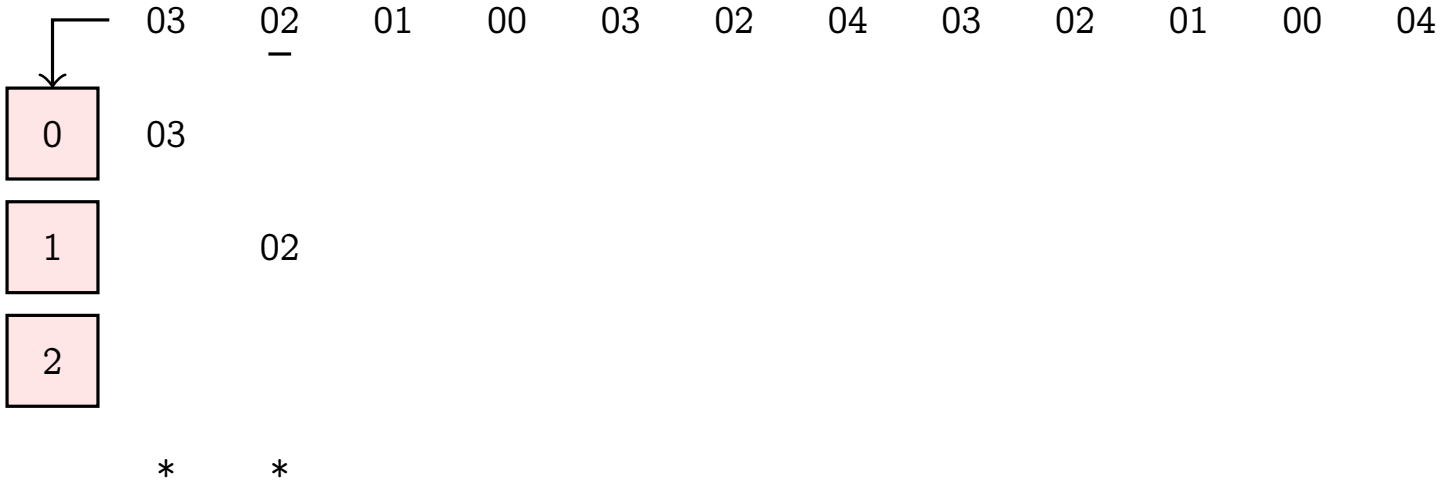


Algorithmes de type FIFO

Sur certaines instances...

Plus de cadres $\longrightarrow$ plus de défauts!

Exemple: avec 3 cadres

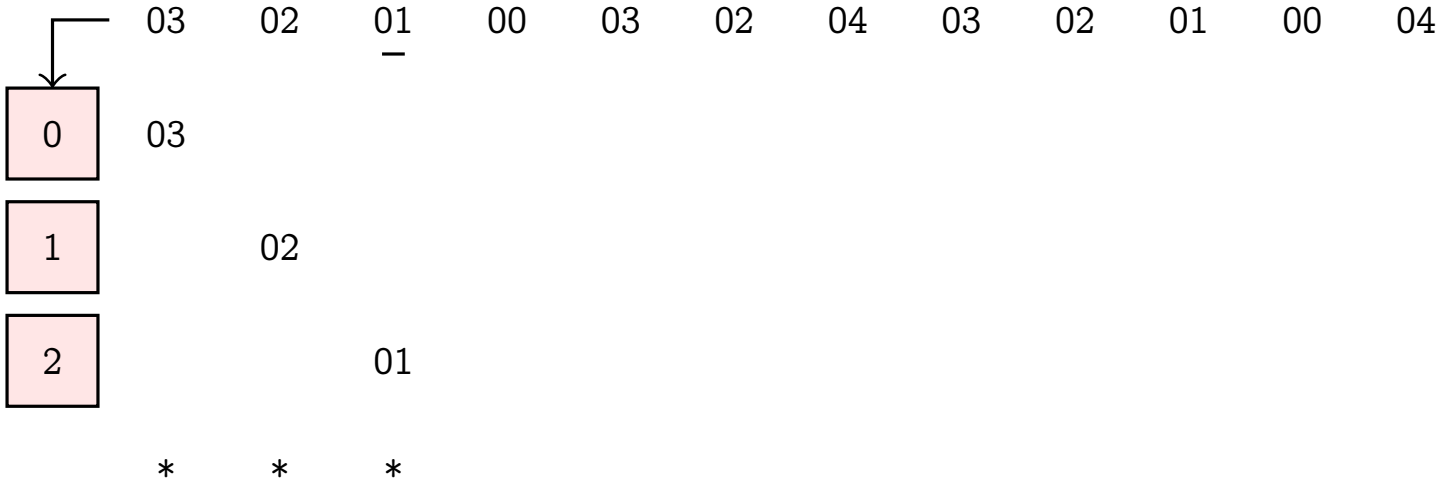


Algorithmes de type FIFO

Sur certaines instances...

Plus de cadres $\longrightarrow$ plus de défauts!

Exemple: avec 3 cadres

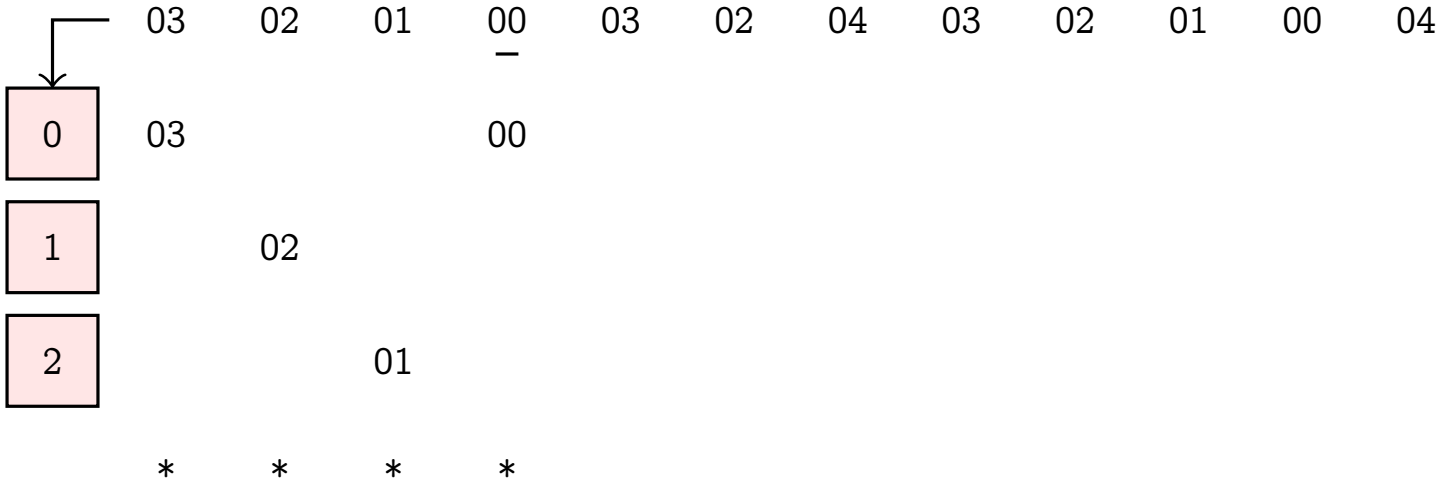


Algorithmes de type FIFO

Sur certaines instances...

Plus de cadres $\longrightarrow$ plus de défauts!

Exemple: avec 3 cadres

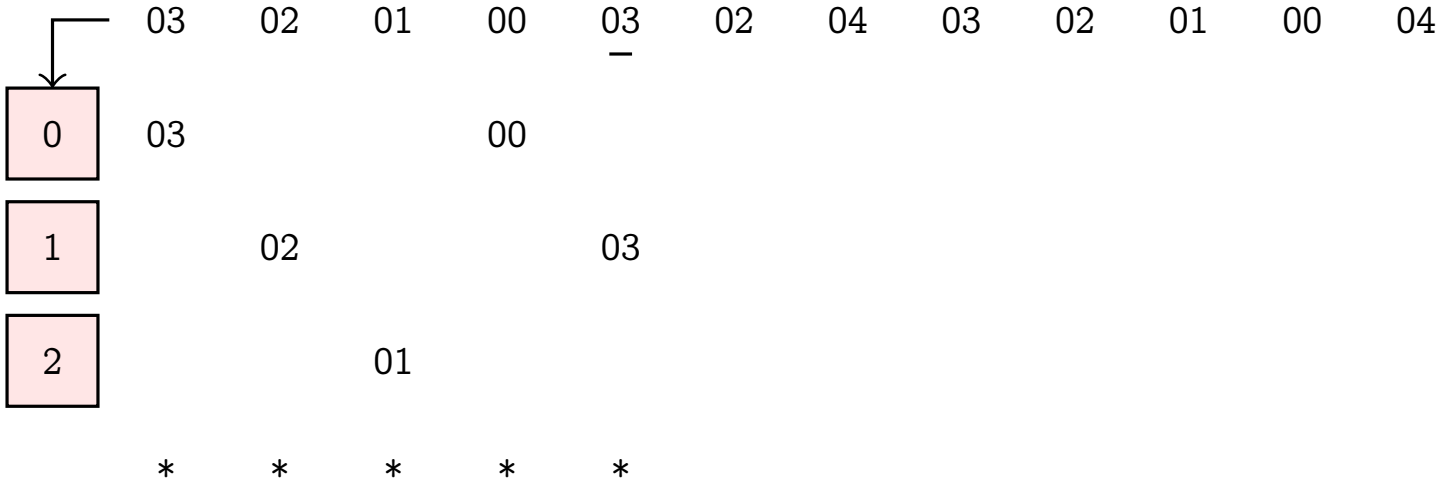


Algorithmes de type FIFO

Sur certaines instances...

Plus de cadres $\longrightarrow$ plus de défauts!

Exemple: avec 3 cadres

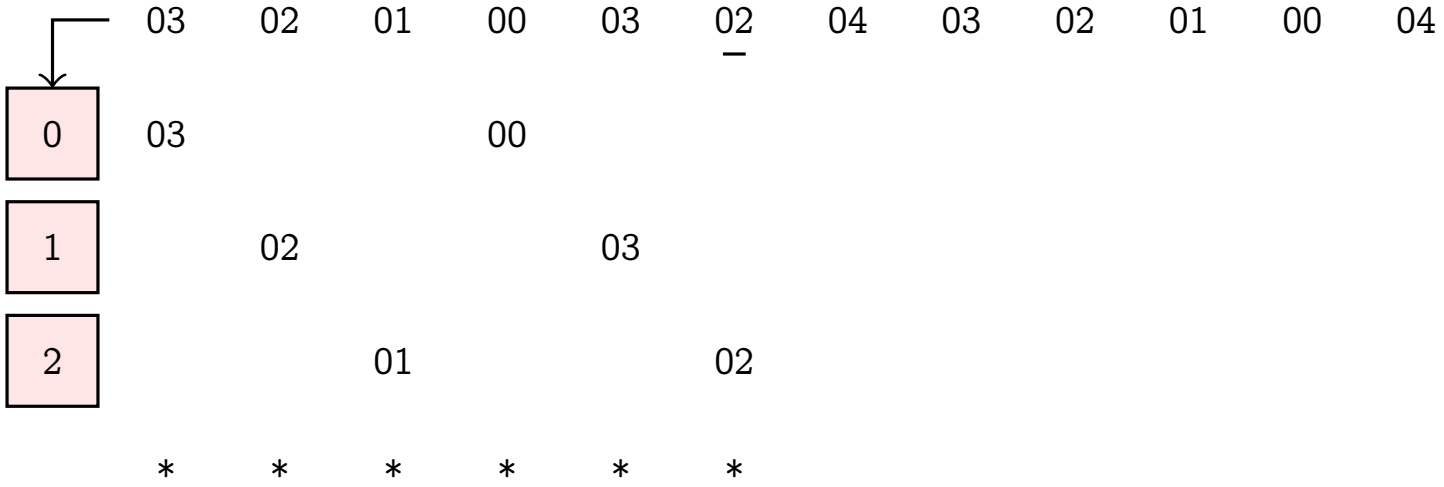


Algorithmes de type FIFO

Sur certaines instances...

Plus de cadres $\longrightarrow$ plus de défauts!

Exemple: avec 3 cadres

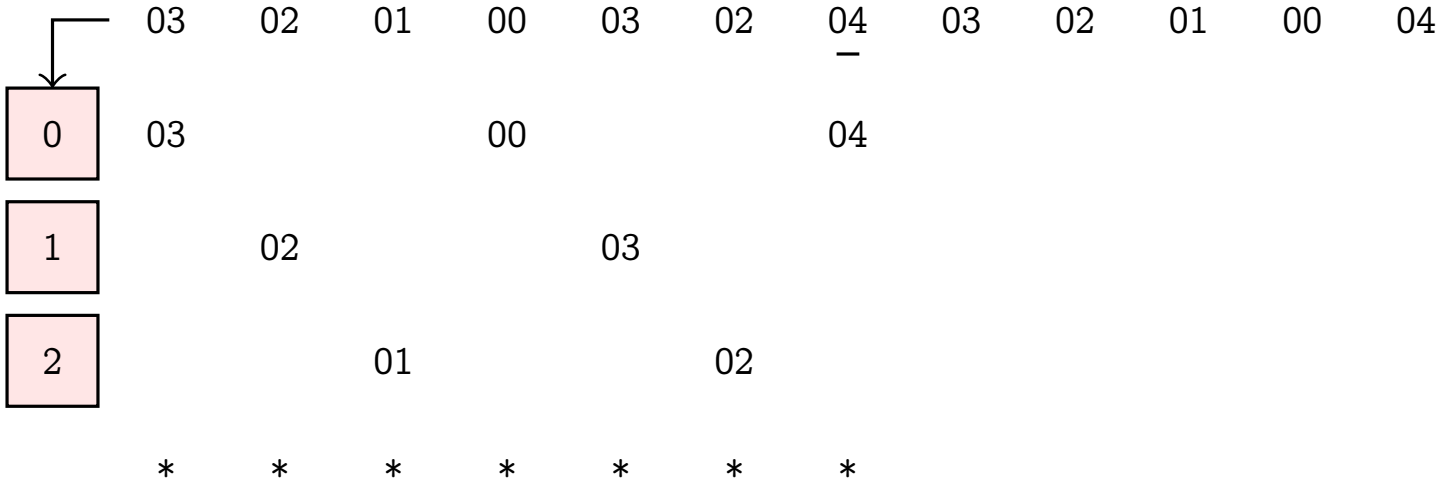


Algorithmes de type FIFO

Sur certaines instances...

Plus de cadres $\longrightarrow$ plus de défauts!

Exemple: avec 3 cadres

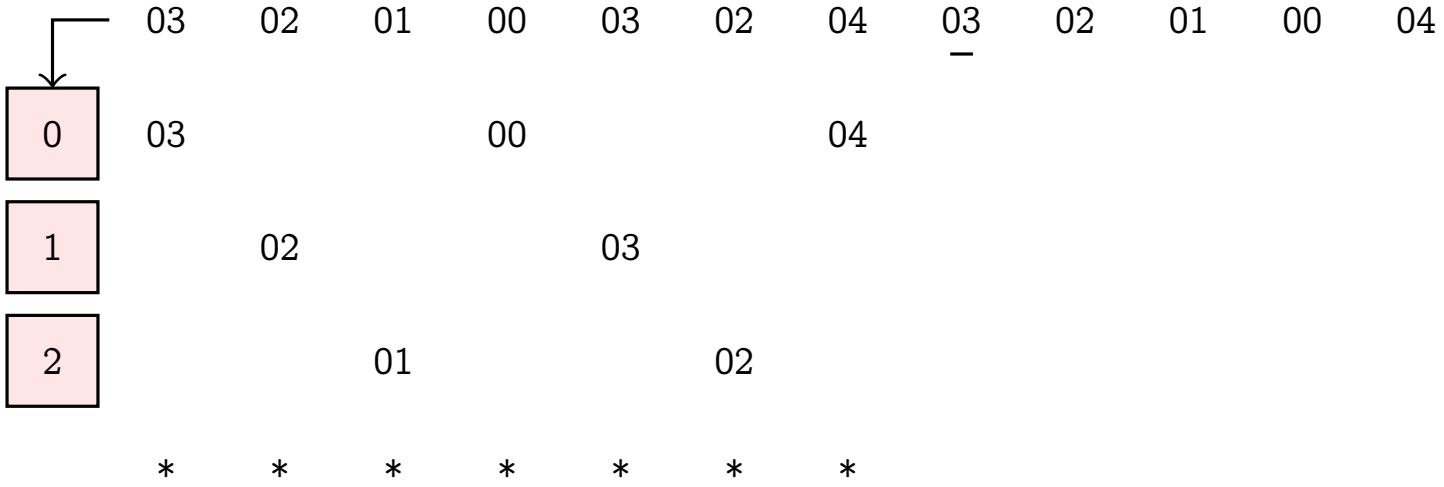


Algorithmes de type FIFO

Sur certaines instances...

Plus de cadres $\longrightarrow$ plus de défauts!

Exemple: avec 3 cadres

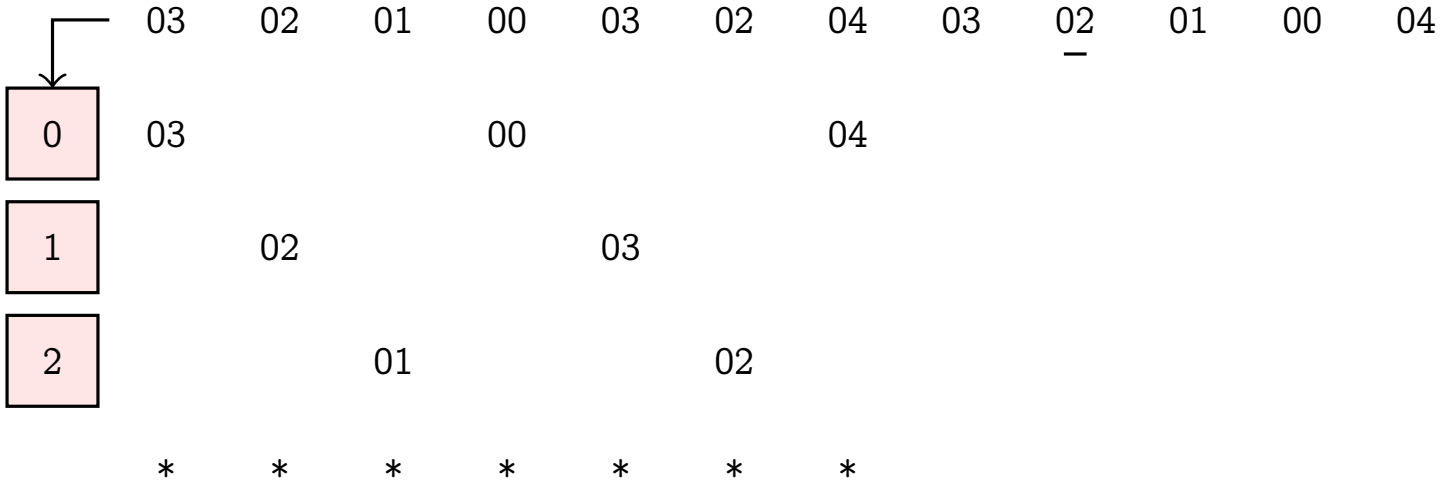


Algorithmes de type FIFO

Sur certaines instances...

Plus de cadres $\longrightarrow$ plus de défauts!

Exemple: avec 3 cadres



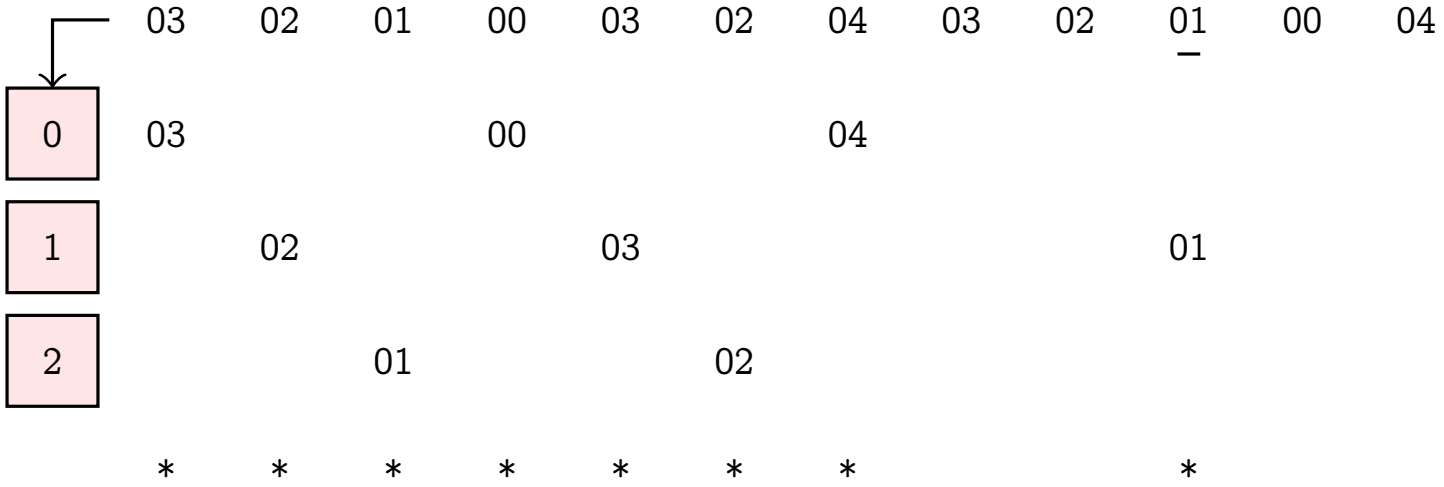
Anomalie de Belady

Algorithmes de type FIFO

Sur certaines instances...

Plus de cadres $\longrightarrow$ plus de défauts!

Exemple: avec 3 cadres

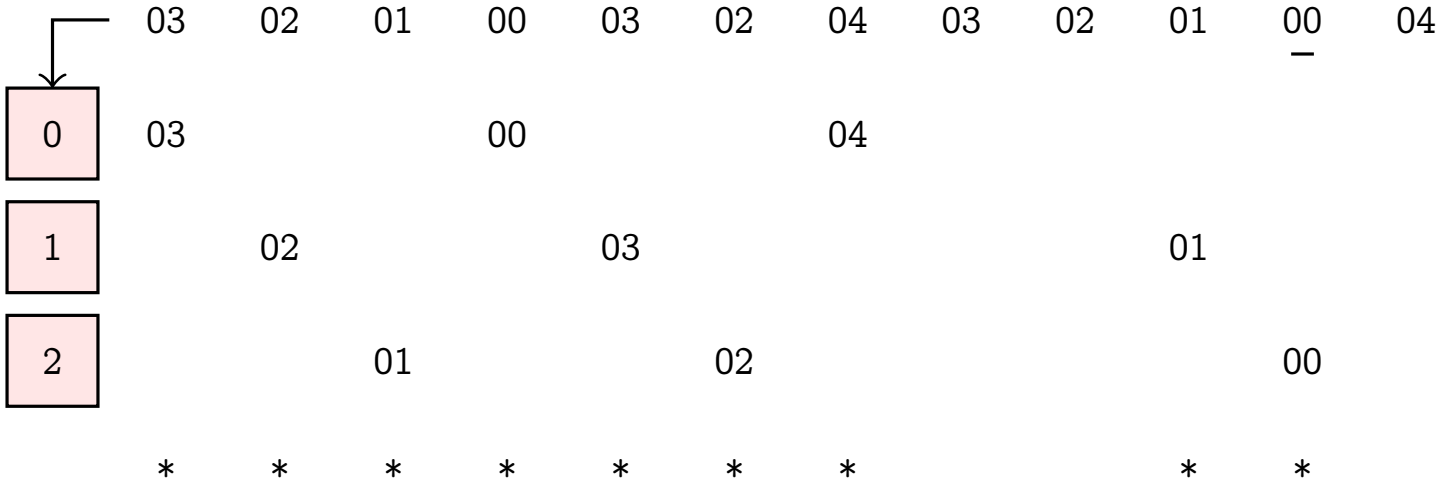


Algorithmes de type FIFO

Sur certaines instances...

Plus de cadres $\longrightarrow$ plus de défauts!

Exemple: avec 3 cadres

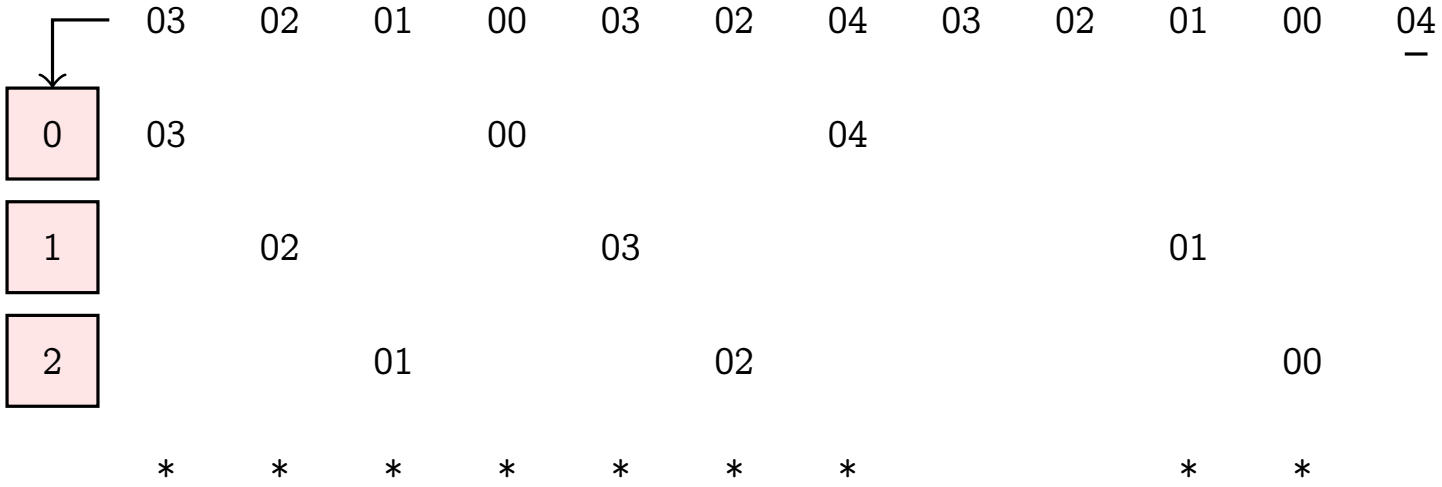


Algorithmes de type FIFO

Sur certaines instances...

Plus de cadres $\longrightarrow$ plus de défauts!

Exemple: avec 3 cadres

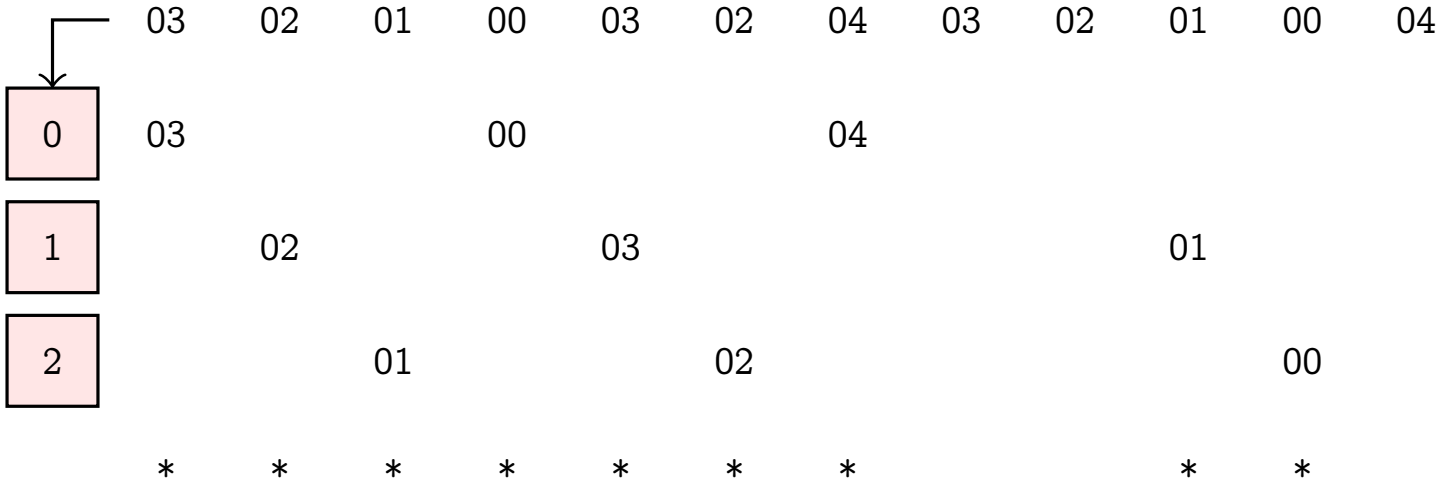


Algorithmes de type FIFO

Sur certaines instances...

Plus de cadres $\longrightarrow$ plus de défauts!

Exemple: avec 3 cadres



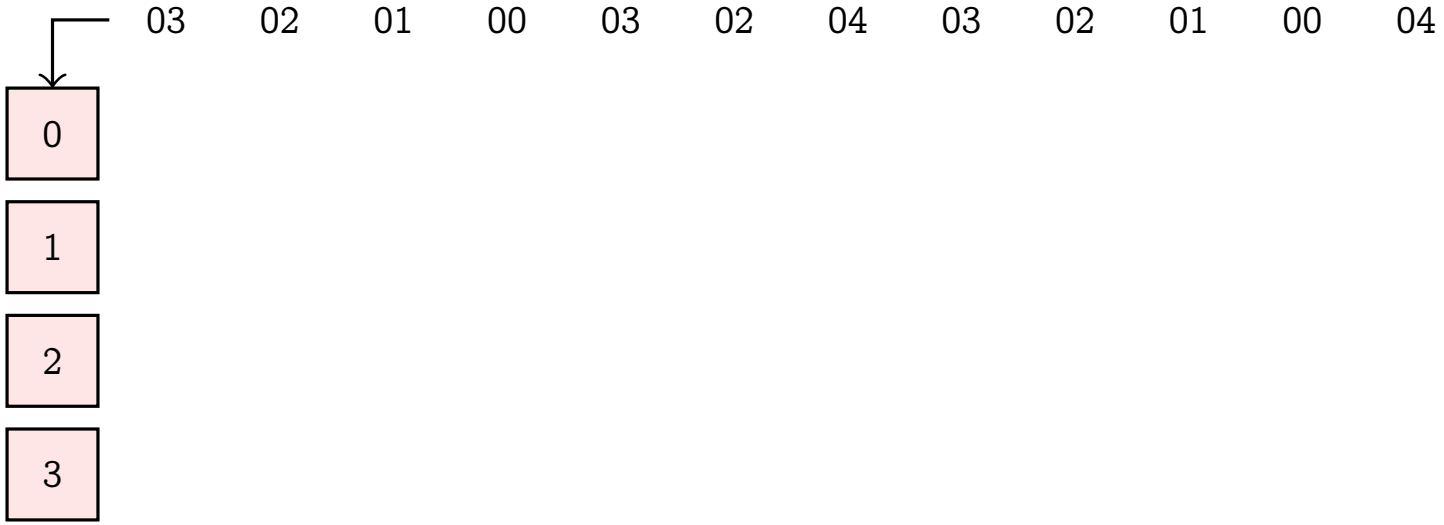
Total: 9 défauts

Algorithmes de type FIFO

Sur certaines instances...

Plus de cadres $\rightarrow$ plus de défauts!

Exemple: avec 4 cadres

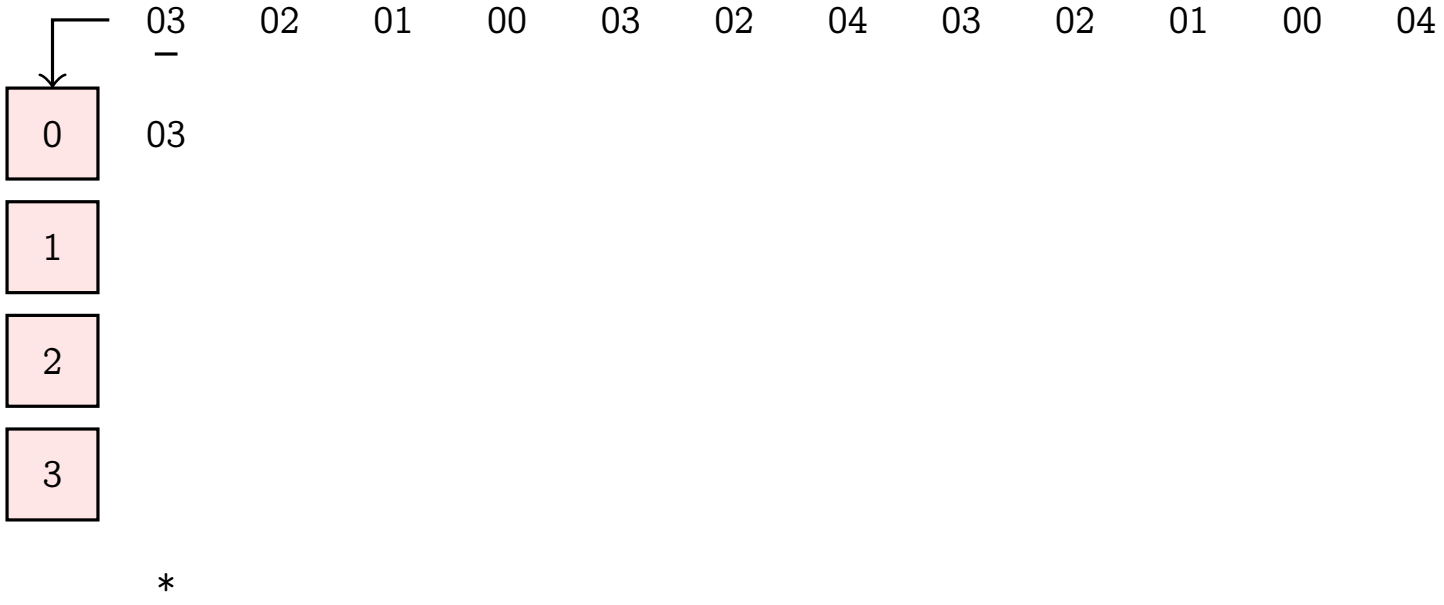


Algorithmes de type FIFO

Sur certaines instances...

Plus de cadres $\longrightarrow$ plus de défauts!

Exemple: avec 4 cadres

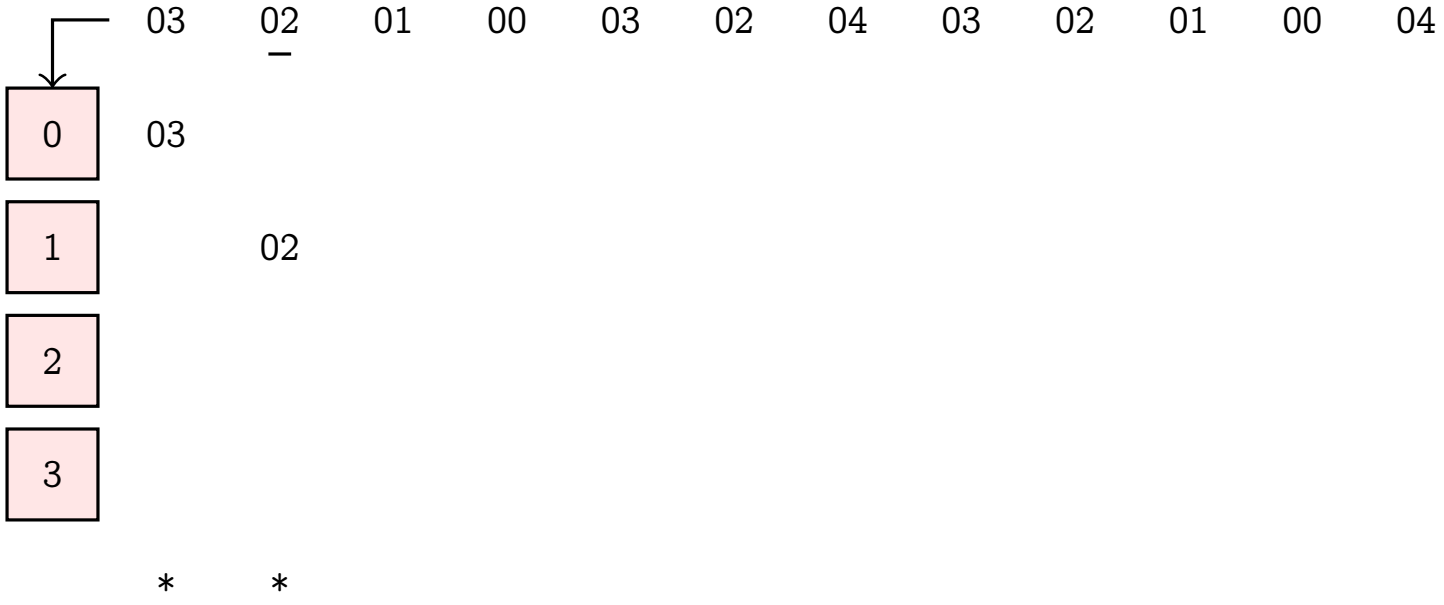


Algorithmes de type FIFO

Sur certaines instances...

Plus de cadres $\longrightarrow$ plus de défauts!

Exemple: avec 4 cadres

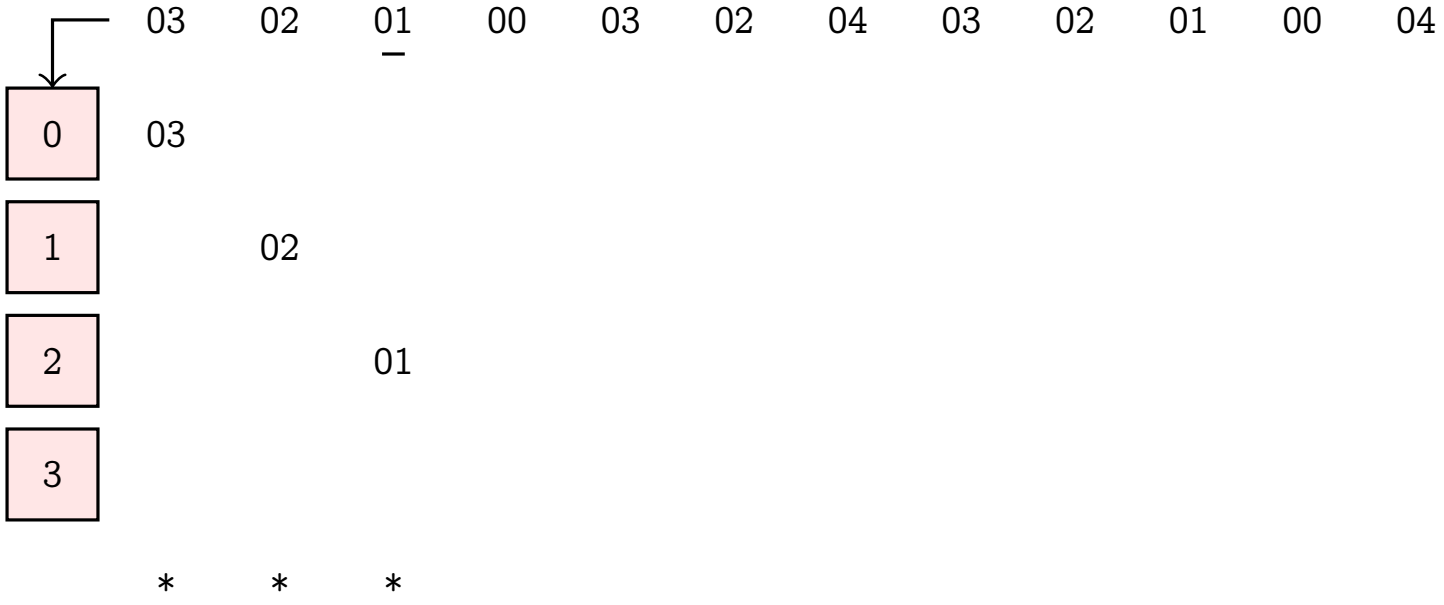


Algorithmes de type FIFO

Sur certaines instances...

Plus de cadres $\longrightarrow$ plus de défauts!

Exemple: avec 4 cadres

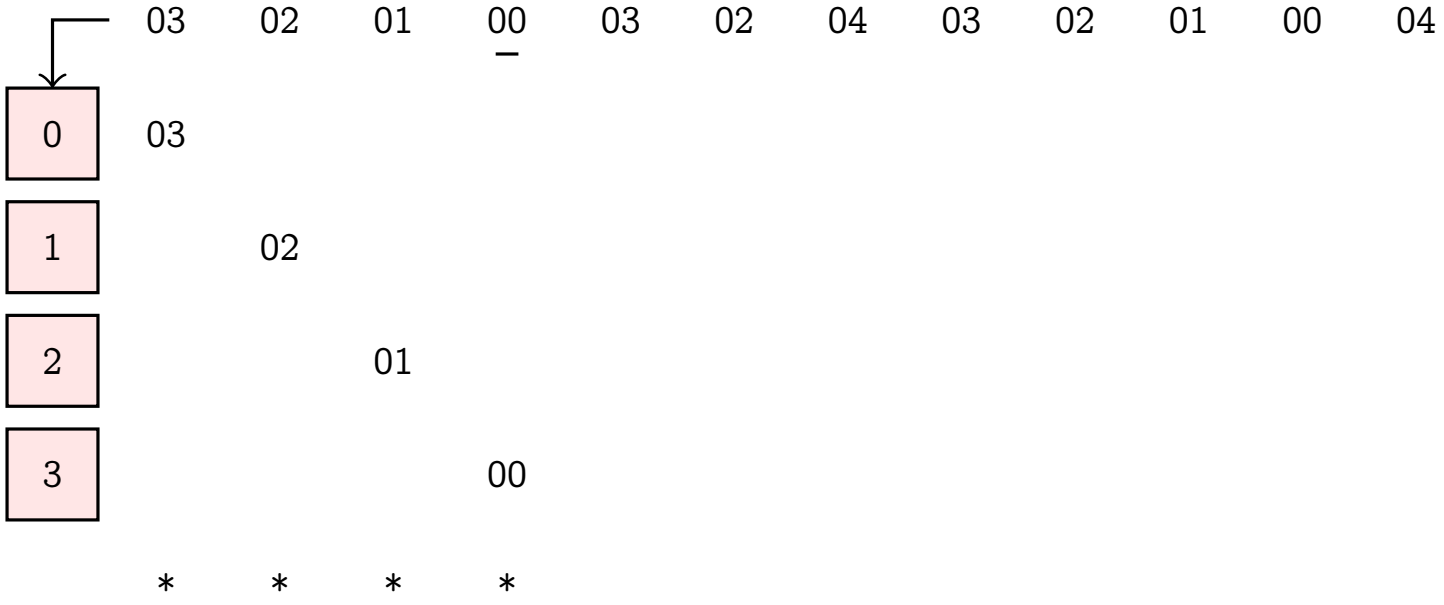


Algorithmes de type FIFO

Sur certaines instances...

Plus de cadres $\longrightarrow$ plus de défauts!

Exemple: avec 4 cadres

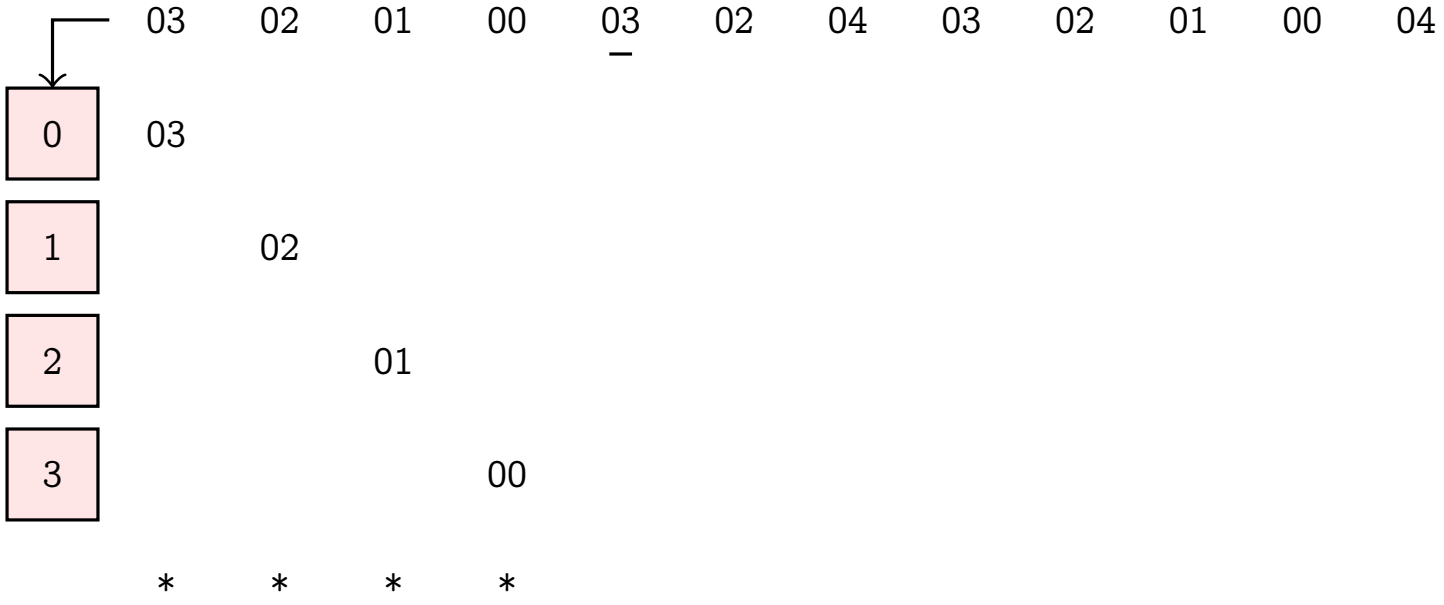


Algorithmes de type FIFO

Sur certaines instances...

Plus de cadres $\longrightarrow$ plus de défauts!

Exemple: avec 4 cadres

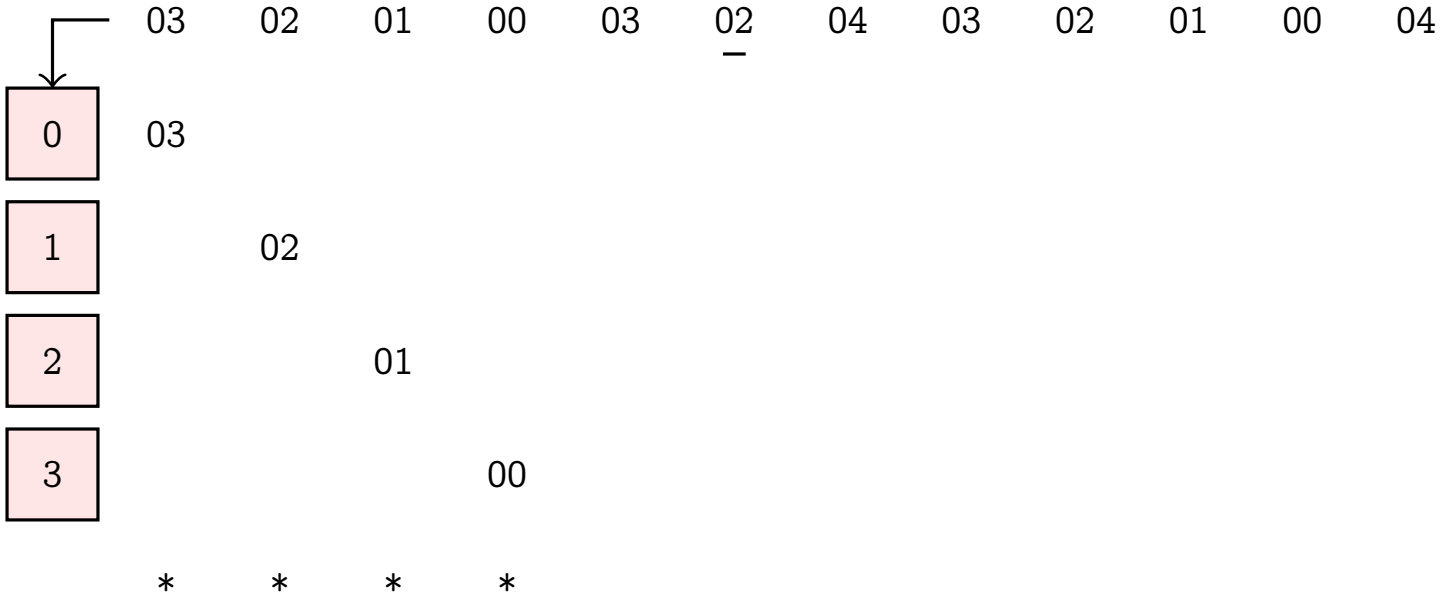


Algorithmes de type FIFO

Sur certaines instances...

Plus de cadres $\longrightarrow$ plus de défauts!

Exemple: avec 4 cadres

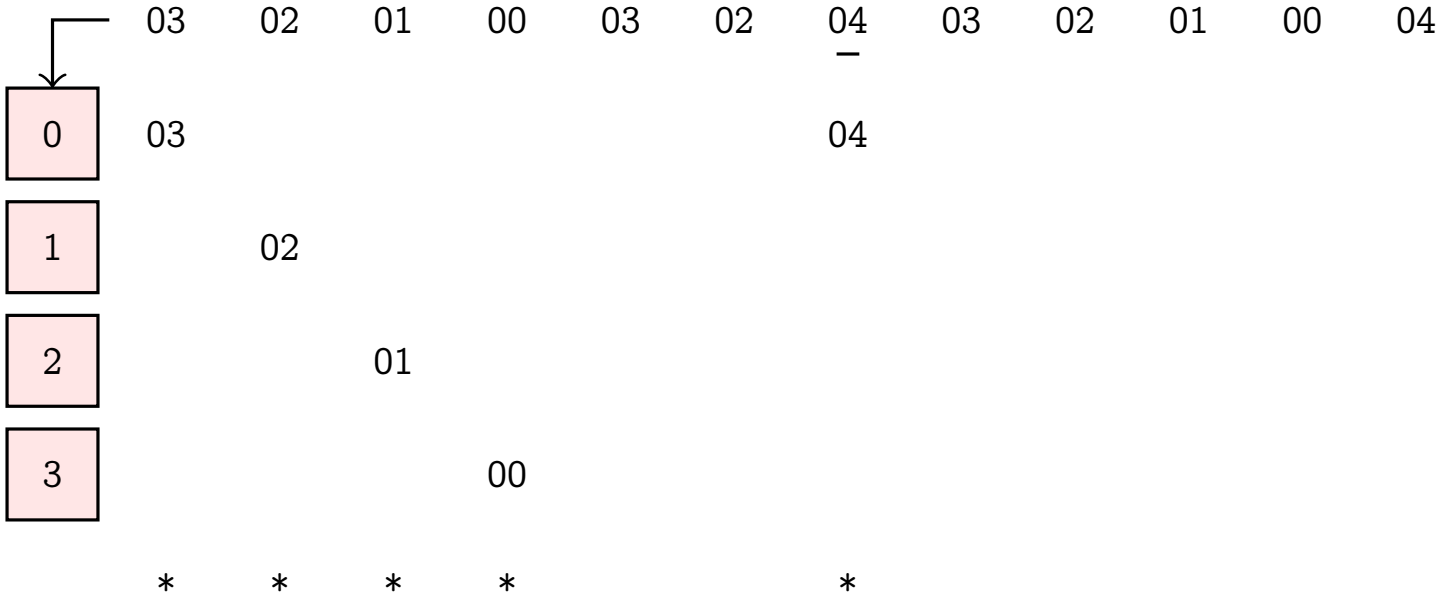


Algorithmes de type FIFO

Sur certaines instances...

Plus de cadres $\longrightarrow$ plus de défauts!

Exemple: avec 4 cadres

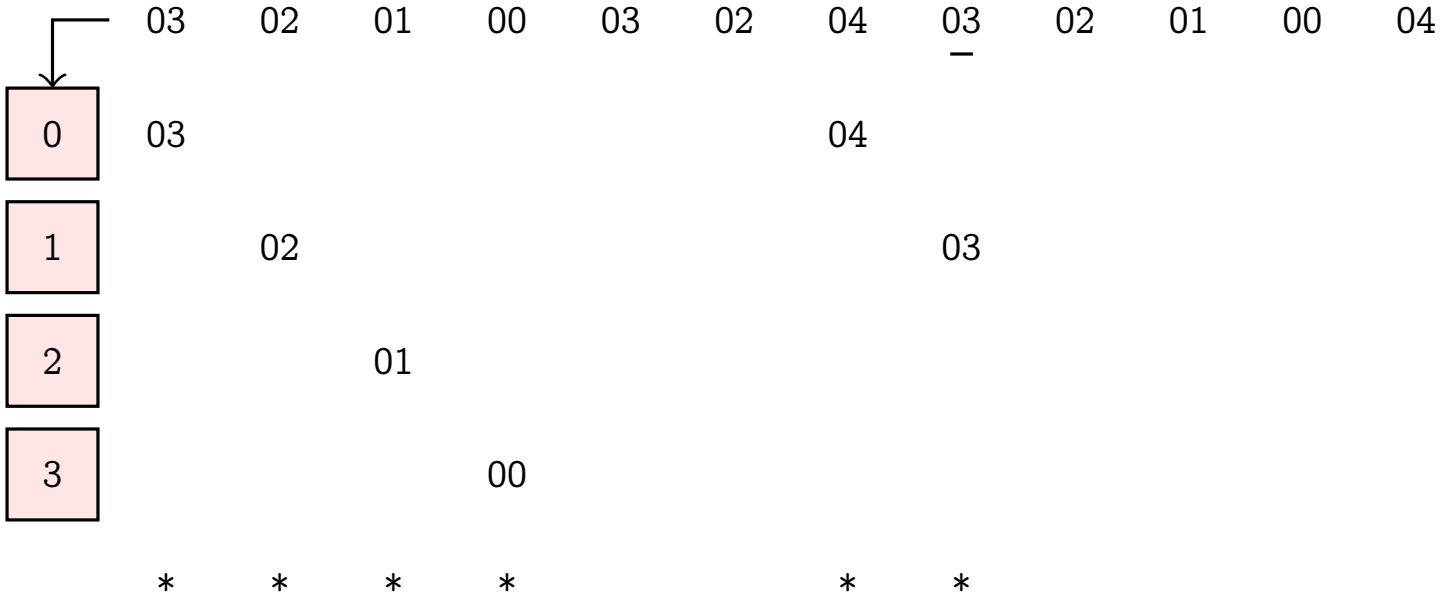


Algorithmes de type FIFO

Sur certaines instances...

Plus de cadres $\longrightarrow$ plus de défauts!

Exemple: avec 4 cadres

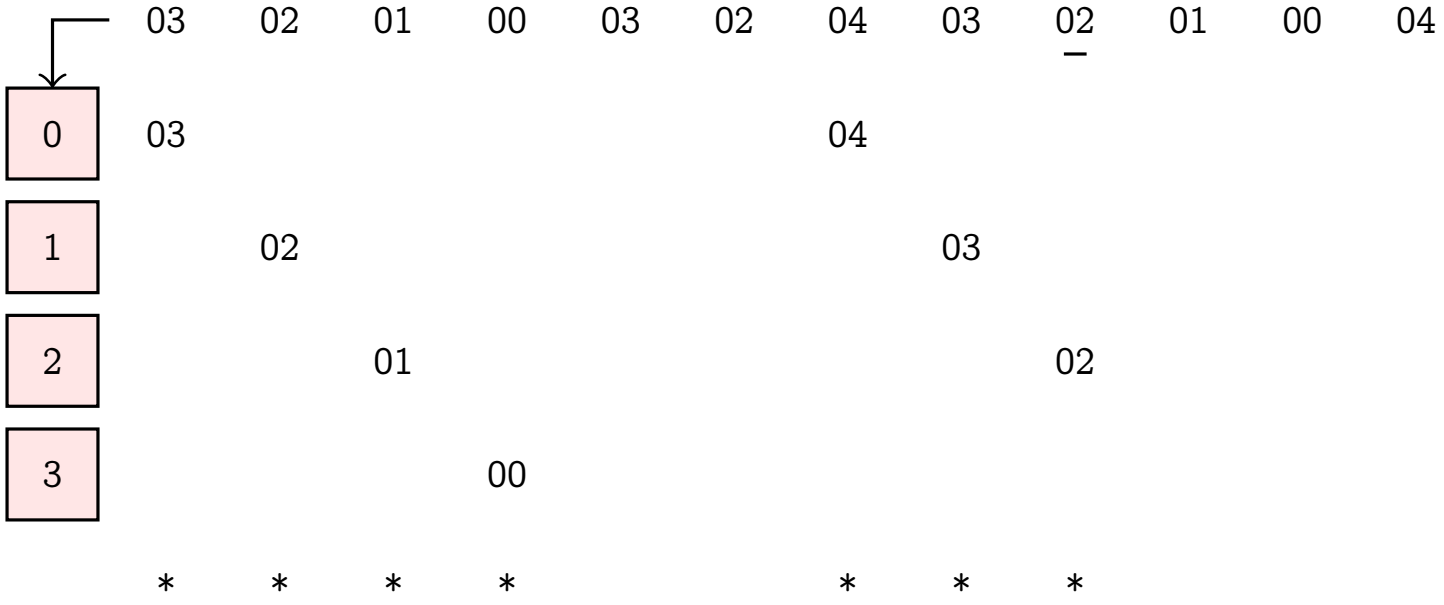


Algorithmes de type FIFO

Sur certaines instances...

Plus de cadres $\longrightarrow$ plus de défauts!

Exemple: avec 4 cadres

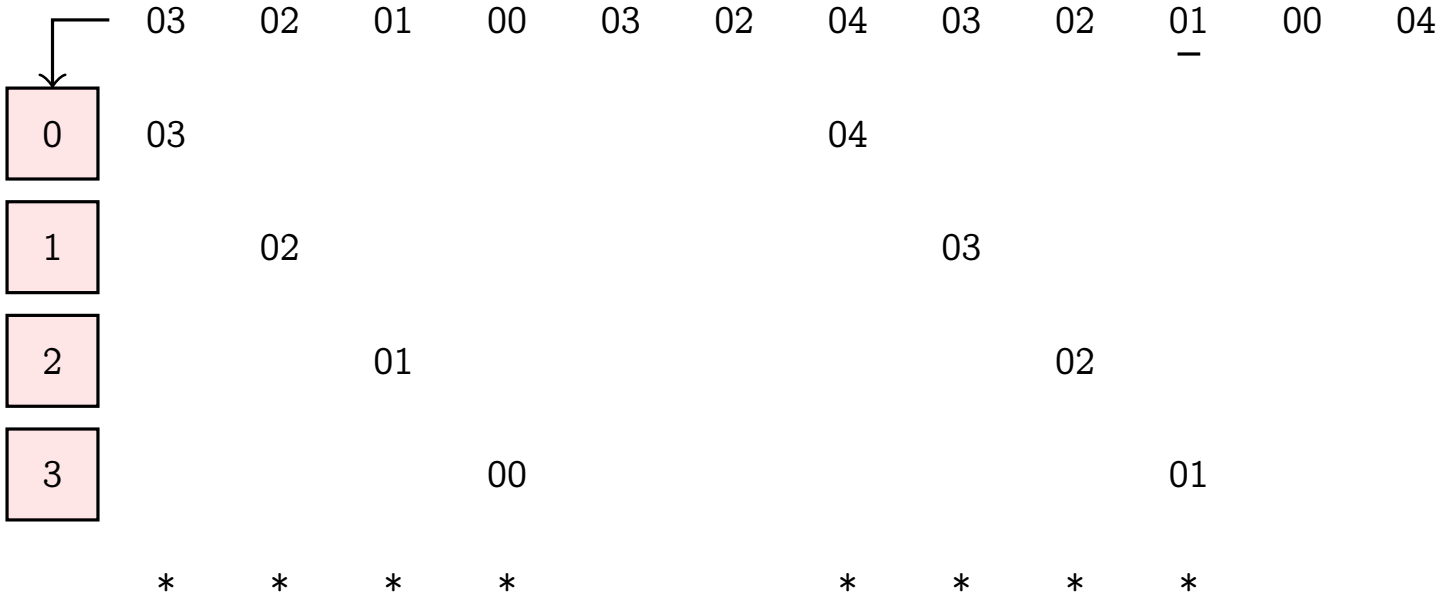


Algorithmes de type FIFO

Sur certaines instances...

Plus de cadres $\longrightarrow$ plus de défauts!

Exemple: avec 4 cadres

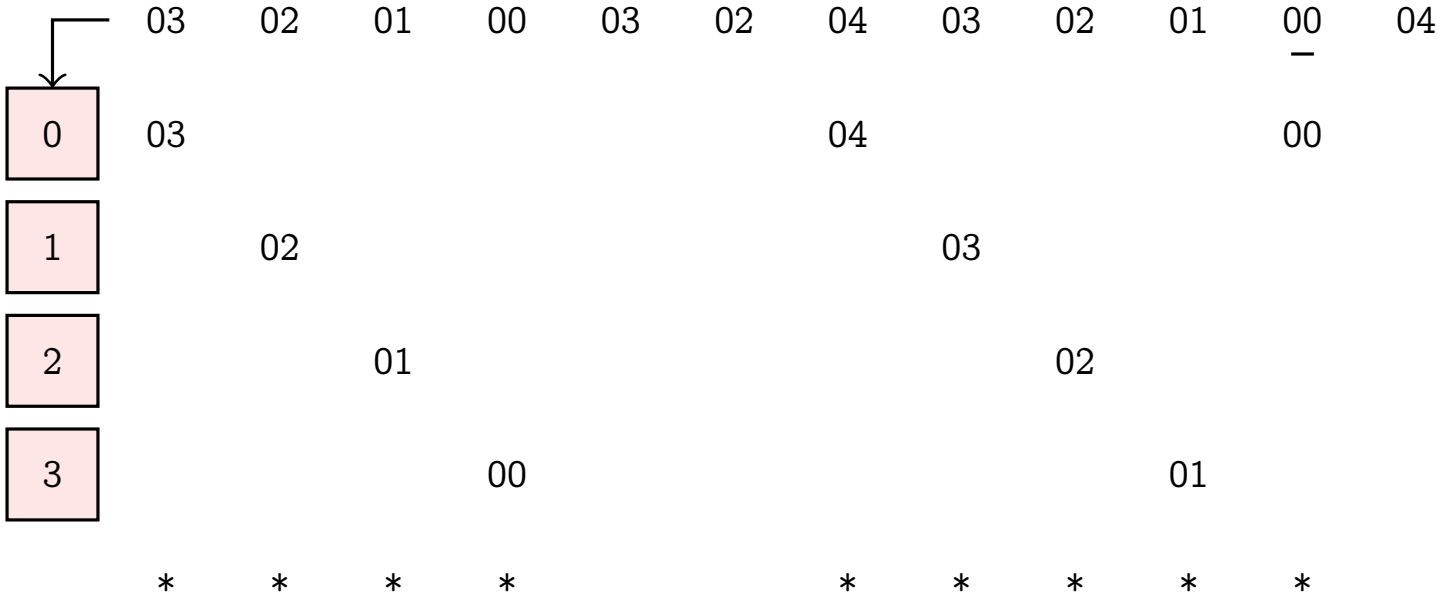


Algorithmes de type FIFO

Sur certaines instances...

Plus de cadres $\longrightarrow$ plus de défauts!

Exemple: avec 4 cadres

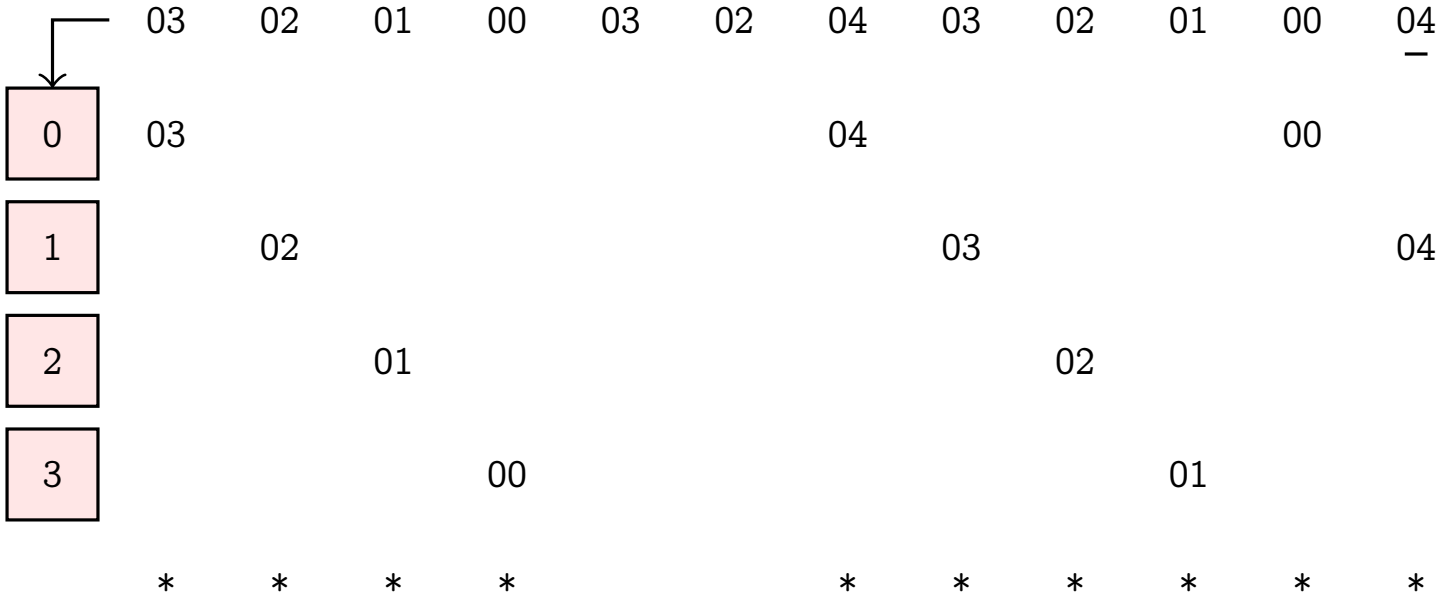


Algorithmes de type FIFO

Sur certaines instances...

Plus de cadres $\longrightarrow$ plus de défauts!

Exemple: avec 4 cadres

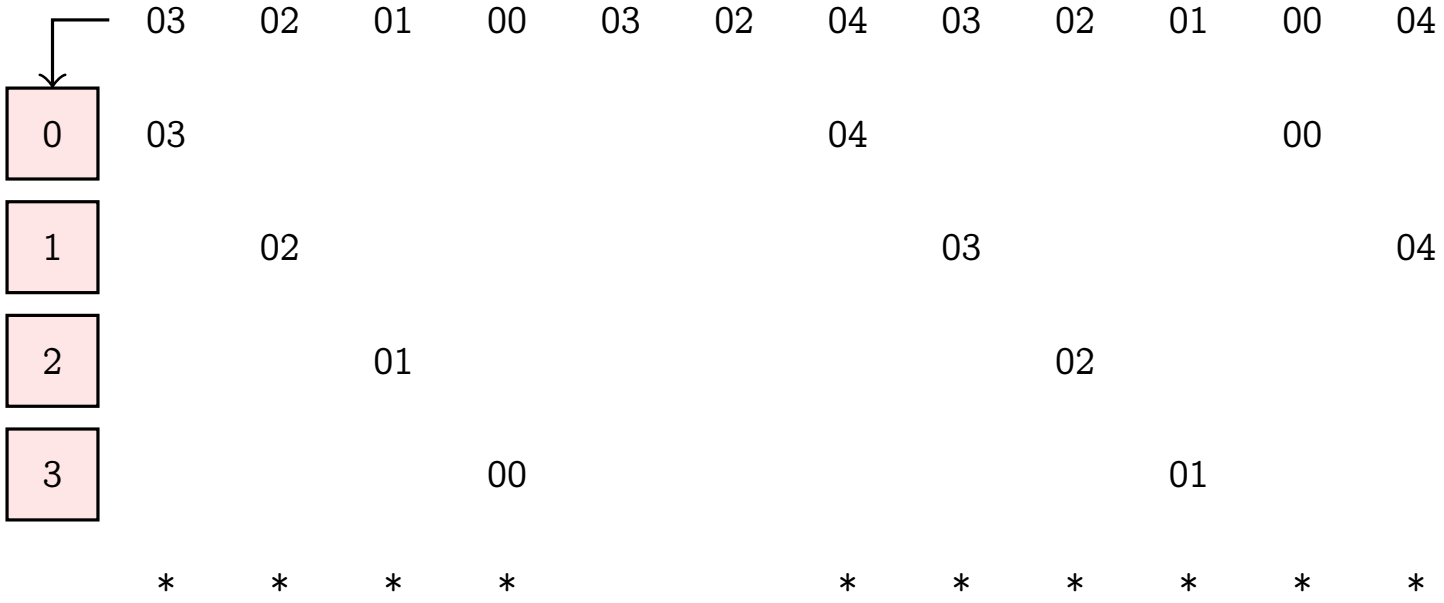


Algorithmes de type FIFO

Sur certaines instances...

Plus de cadres $\longrightarrow$ plus de défauts!

Exemple: avec 4 cadres



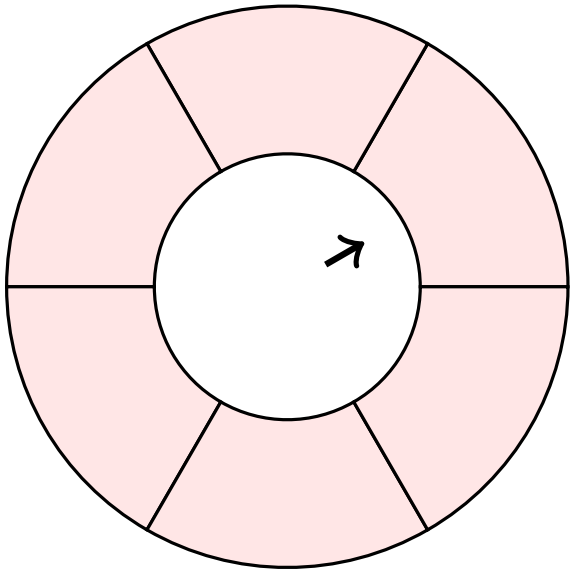
Total: 10 défauts

Seconde chance

Principe

- Bit de 2nd chance mis à 0 lors d'un acces
- Lorsqu'on a besoin de libérer un cadre:
  - Si bit=0, mettre le bit à 1 et passer
  - Sinon utiliser le cadre (et remettre le bit à 0)

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

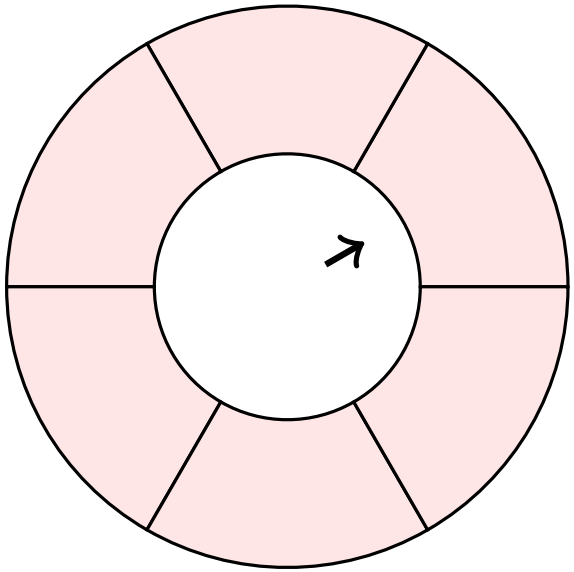


Seconde chance

Principe

- Bit de 2nd chance mis à 0 lors d'un acces
- Lorsqu'on a besoin de libérer un cadre:
  - Si bit=0, mettre le bit à 1 et passer
  - Sinon utiliser le cadre (et remettre le bit à 0)

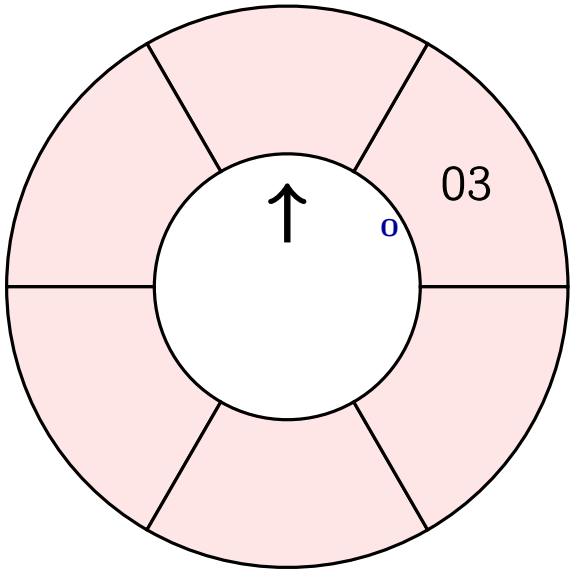
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



Principe

- Bit de 2nd chance mis à 0 lors d'un acces
- Lorsqu'on a besoin de libérer un cadre:
  - Si bit=0, mettre le bit à 1 et passer
  - Sinon utiliser le cadre (et remettre le bit à 0)

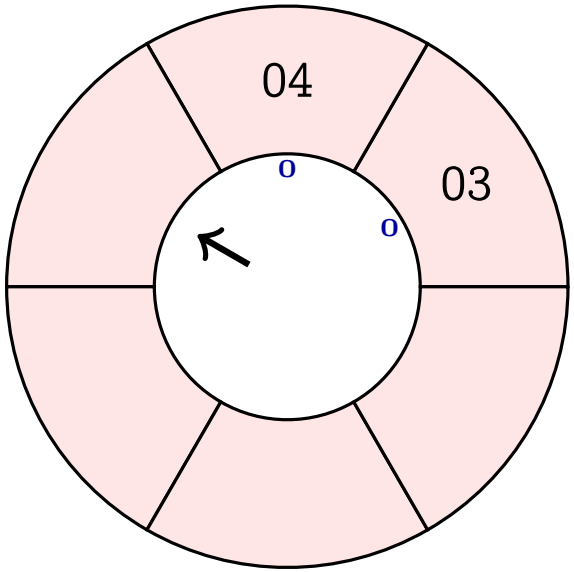
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



Principe

- Bit de 2nd chance mis à 0 lors d'un acces
- Lorsqu'on a besoin de libérer un cadre:
  - Si bit=0, mettre le bit à 1 et passer
  - Sinon utiliser le cadre (et remettre le bit à 0)

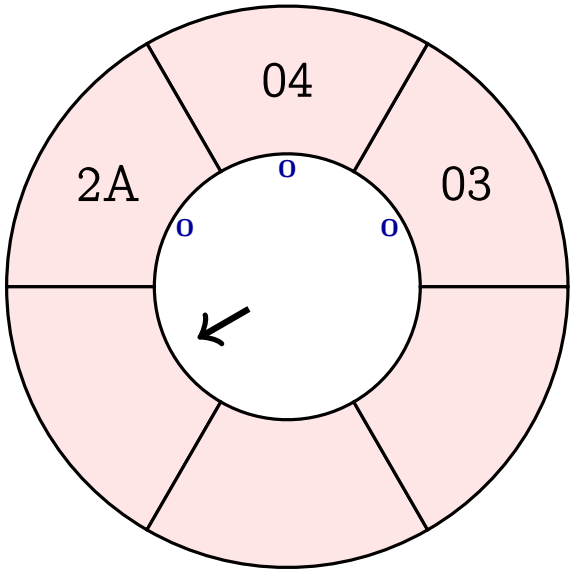
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



Principe

- Bit de 2nd chance mis à 0 lors d'un acces
- Lorsqu'on a besoin de libérer un cadre:
  - Si bit=0, mettre le bit à 1 et passer
  - Sinon utiliser le cadre (et remettre le bit à 0)

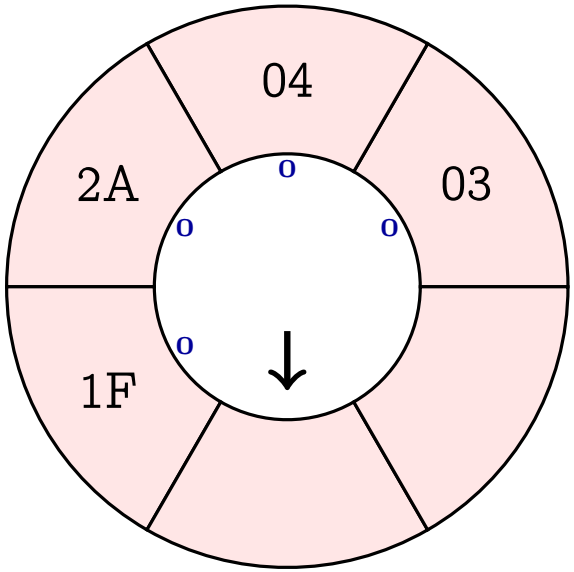
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



Principe

- Bit de 2nd chance mis à 0 lors d'un acces
- Lorsqu'on a besoin de libérer un cadre:
  - Si bit=0, mettre le bit à 1 et passer
  - Sinon utiliser le cadre (et remettre le bit à 0)

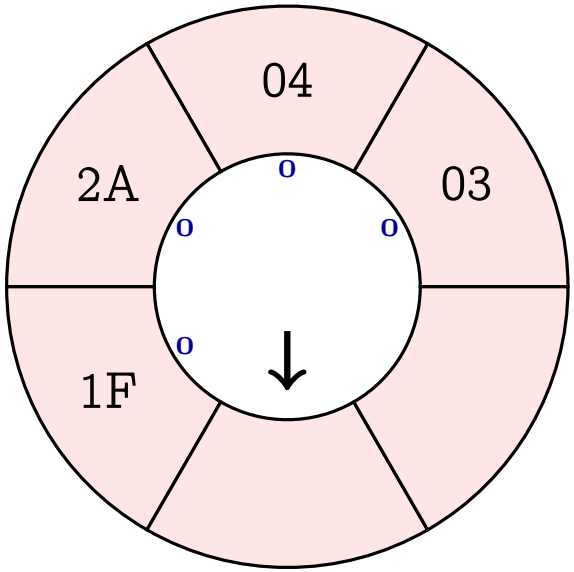
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



Principe

- Bit de 2nd chance mis à 0 lors d'un acces
- Lorsqu'on a besoin de libérer un cadre:
  - Si bit=0, mettre le bit à 1 et passer
  - Sinon utiliser le cadre (et remettre le bit à 0)

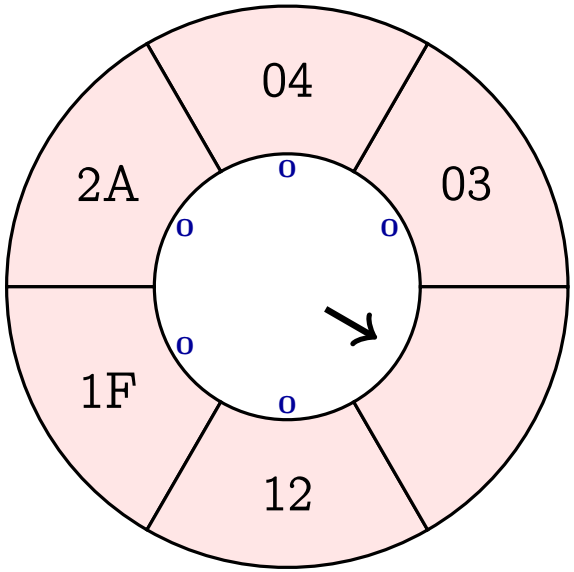
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



Principe

- Bit de 2nd chance mis à 0 lors d'un acces
- Lorsqu'on a besoin de libérer un cadre:
  - Si bit=0, mettre le bit à 1 et passer
  - Sinon utiliser le cadre (et remettre le bit à 0)

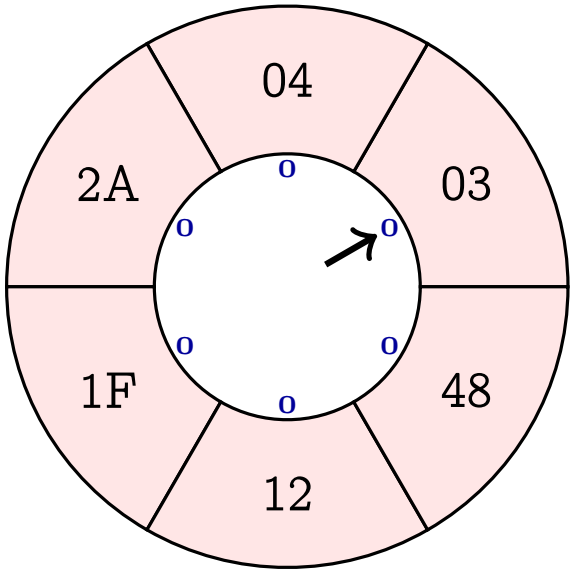
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



Principe

- Bit de 2nd chance mis à 0 lors d'un acces
- Lorsqu'on a besoin de libérer un cadre:
  - Si bit=0, mettre le bit à 1 et passer
  - Sinon utiliser le cadre (et remettre le bit à 0)

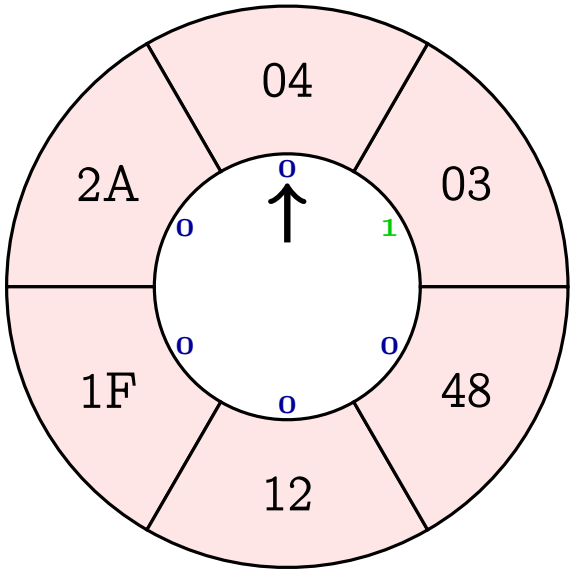
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



Principe

- Bit de 2nd chance mis à 0 lors d'un acces
- Lorsqu'on a besoin de libérer un cadre:
  - Si bit=0, mettre le bit à 1 et passer
  - Sinon utiliser le cadre (et remettre le bit à 0)

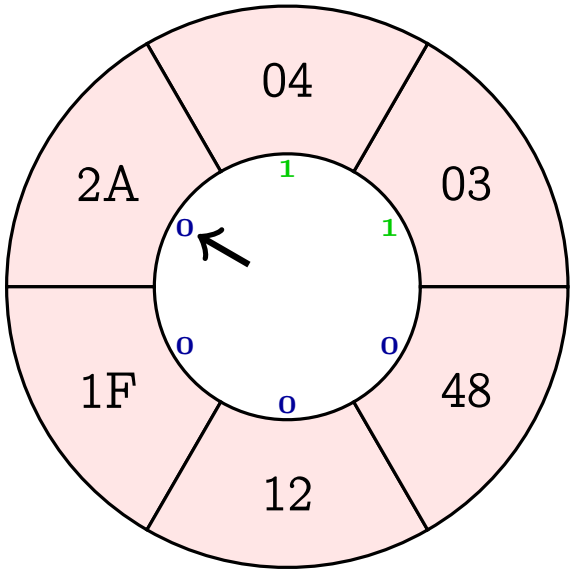
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



Principe

- Bit de 2nd chance mis à 0 lors d'un acces
- Lorsqu'on a besoin de libérer un cadre:
  - Si bit=0, mettre le bit à 1 et passer
  - Sinon utiliser le cadre (et remettre le bit à 0)

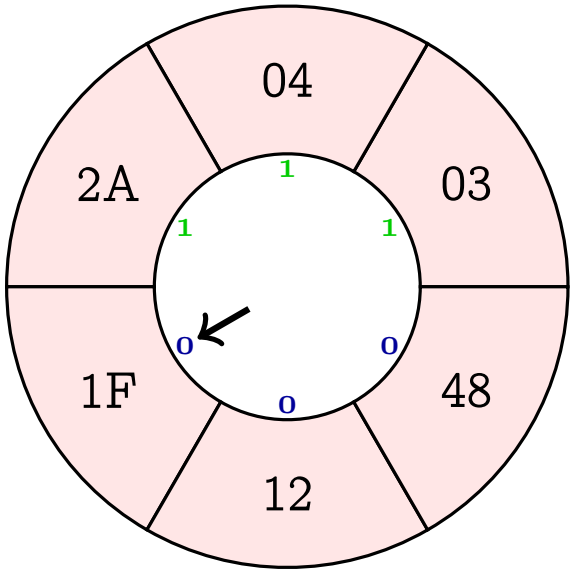
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



Principe

- Bit de 2nd chance mis à 0 lors d'un acces
- Lorsqu'on a besoin de libérer un cadre:
  - Si bit=0, mettre le bit à 1 et passer
  - Sinon utiliser le cadre (et remettre le bit à 0)

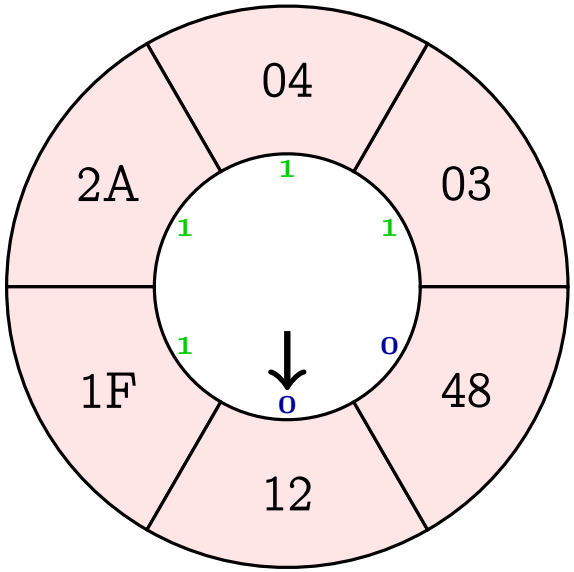
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



Principe

- Bit de 2nd chance mis à 0 lors d'un acces
- Lorsqu'on a besoin de libérer un cadre:
  - Si bit=0, mettre le bit à 1 et passer
  - Sinon utiliser le cadre (et remettre le bit à 0)

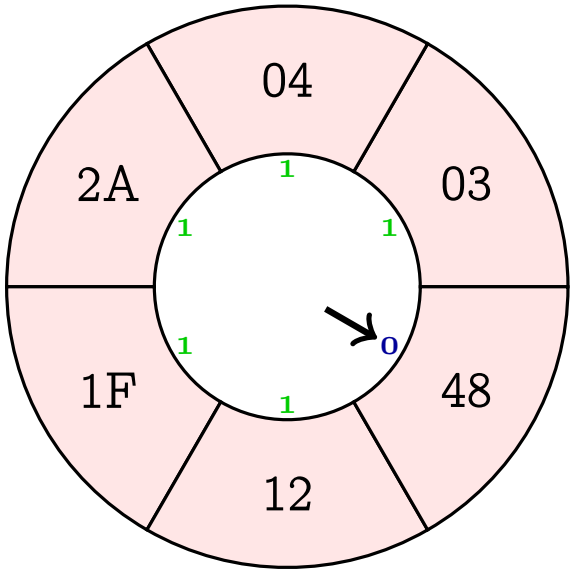
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



Principe

- Bit de 2nd chance mis à 0 lors d'un acces
- Lorsqu'on a besoin de libérer un cadre:
  - Si bit=0, mettre le bit à 1 et passer
  - Sinon utiliser le cadre (et remettre le bit à 0)

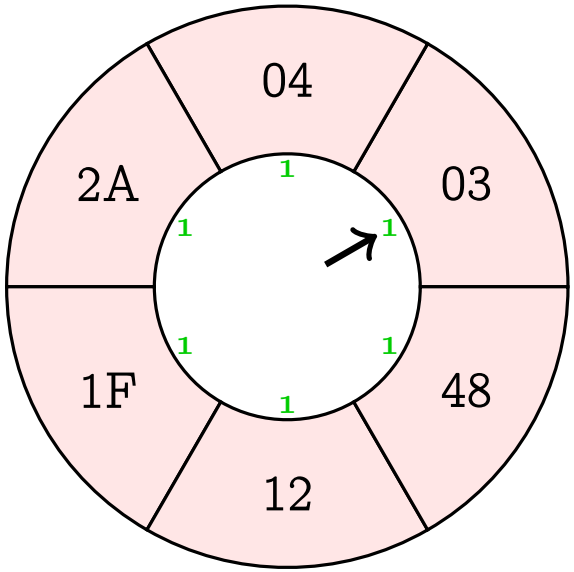
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



Principe

- Bit de 2nd chance mis à 0 lors d'un acces
- Lorsqu'on a besoin de libérer un cadre:
  - Si bit=0, mettre le bit à 1 et passer
  - Sinon utiliser le cadre (et remettre le bit à 0)

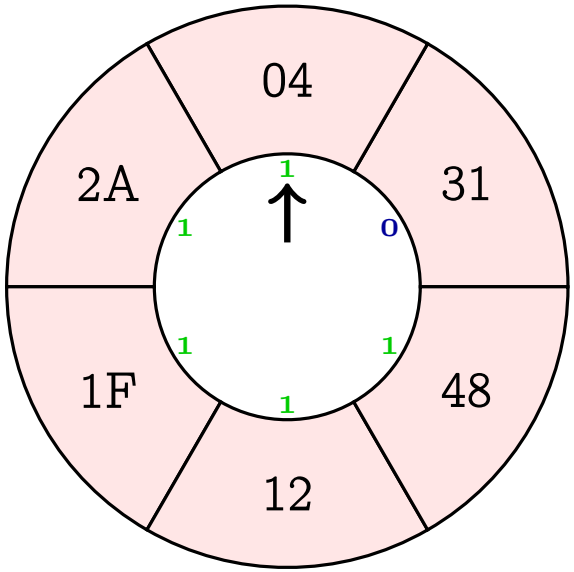
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



Principe

- Bit de 2nd chance mis à 0 lors d'un acces
- Lorsqu'on a besoin de libérer un cadre:
  - Si bit=0, mettre le bit à 1 et passer
  - Sinon utiliser le cadre (et remettre le bit à 0)

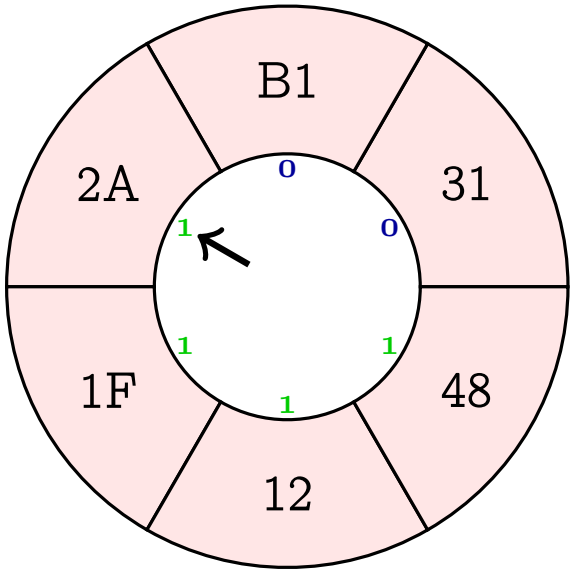
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



Principe

- Bit de 2nd chance mis à 0 lors d'un acces
- Lorsqu'on a besoin de libérer un cadre:
  - Si bit=0, mettre le bit à 1 et passer
  - Sinon utiliser le cadre (et remettre le bit à 0)

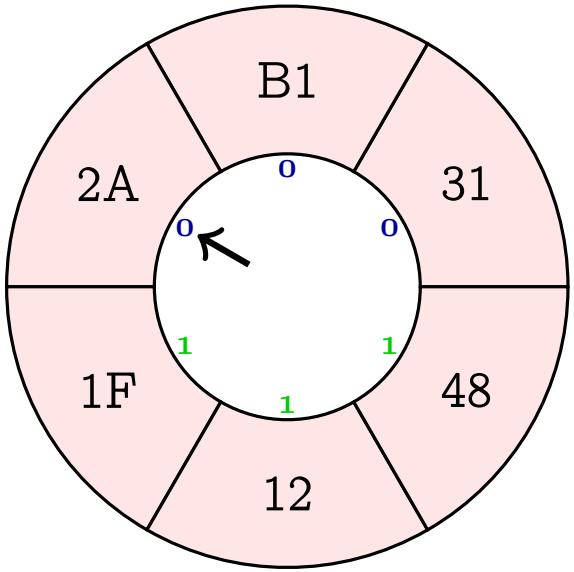
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



Principe

- Bit de 2nd chance mis à 0 lors d'un acces
- Lorsqu'on a besoin de libérer un cadre:
  - Si bit=0, mettre le bit à 1 et passer
  - Sinon utiliser le cadre (et remettre le bit à 0)

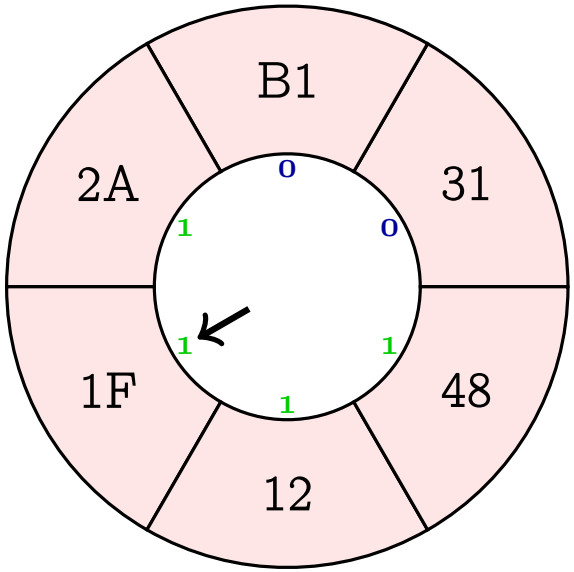
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



Principe

- Bit de 2nd chance mis à 0 lors d'un acces
- Lorsqu'on a besoin de libérer un cadre:
  - Si bit=0, mettre le bit à 1 et passer
  - Sinon utiliser le cadre (et remettre le bit à 0)

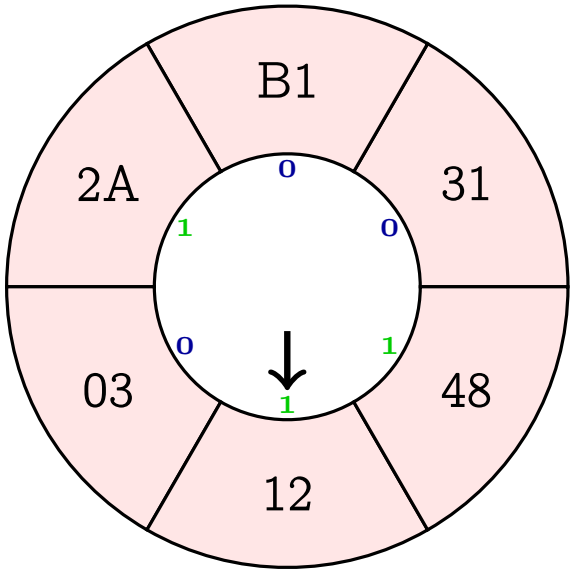
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



Principe

- Bit de 2nd chance mis à 0 lors d'un acces
- Lorsqu'on a besoin de libérer un cadre:
  - Si bit=0, mettre le bit à 1 et passer
  - Sinon utiliser le cadre (et remettre le bit à 0)

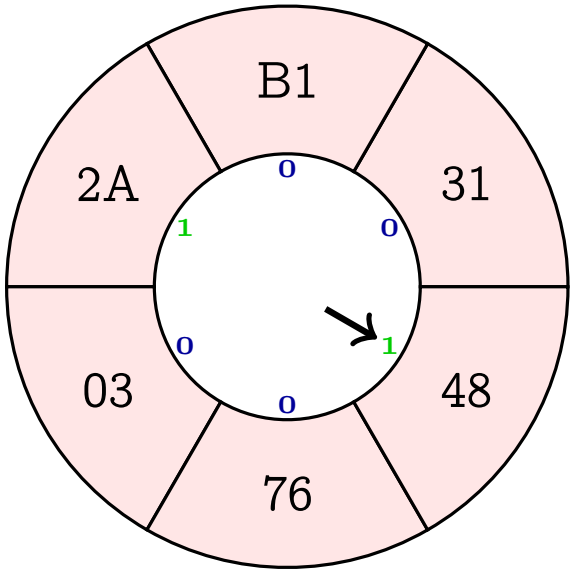
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



Principe

- Bit de 2nd chance mis à 0 lors d'un acces
- Lorsqu'on a besoin de libérer un cadre:
  - Si bit=0, mettre le bit à 1 et passer
  - Sinon utiliser le cadre (et remettre le bit à 0)

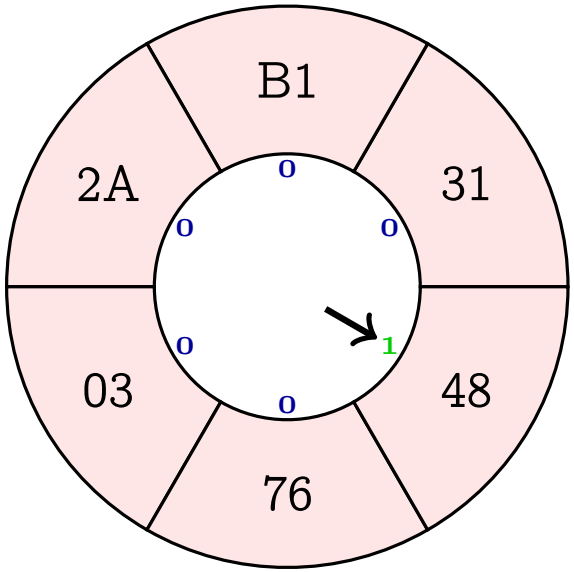
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



Principe

- Bit de 2nd chance mis à 0 lors d'un acces
- Lorsqu'on a besoin de libérer un cadre:
  - Si bit=0, mettre le bit à 1 et passer
  - Sinon utiliser le cadre (et remettre le bit à 0)

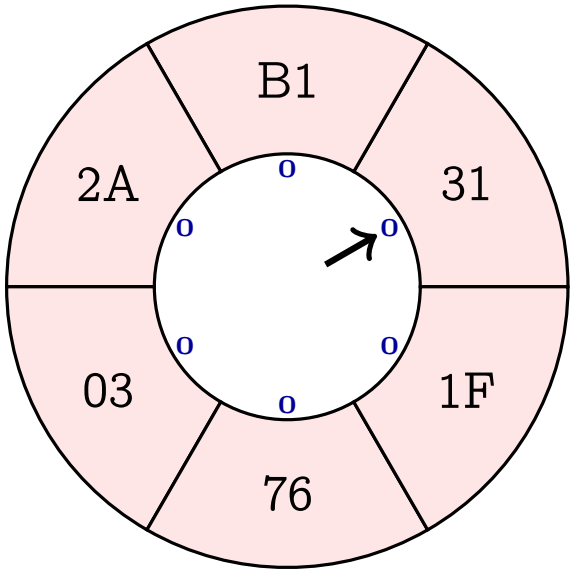
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



Principe

- Bit de 2nd chance mis à 0 lors d'un acces
- Lorsqu'on a besoin de libérer un cadre:
  - Si bit=0, mettre le bit à 1 et passer
  - Sinon utiliser le cadre (et remettre le bit à 0)

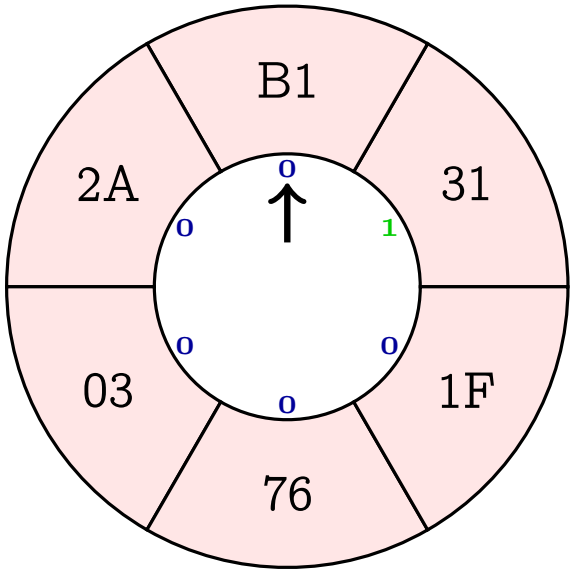
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



Principe

- Bit de 2nd chance mis à 0 lors d'un acces
- Lorsqu'on a besoin de libérer un cadre:
  - Si bit=0, mettre le bit à 1 et passer
  - Sinon utiliser le cadre (et remettre le bit à 0)

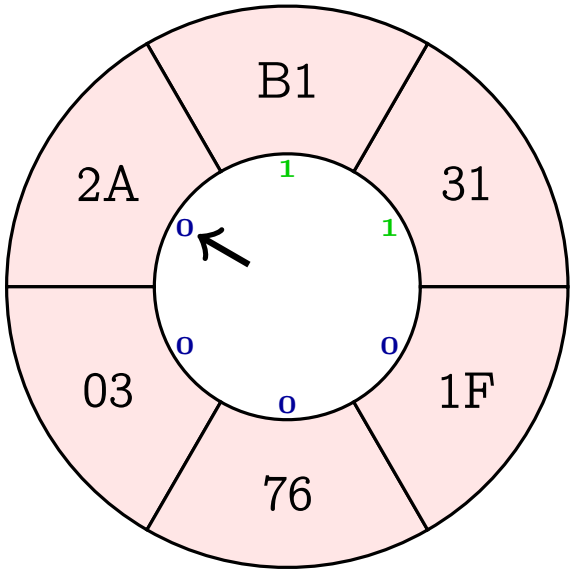
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



Principe

- Bit de 2nd chance mis à 0 lors d'un acces
- Lorsqu'on a besoin de libérer un cadre:
  - Si bit=0, mettre le bit à 1 et passer
  - Sinon utiliser le cadre (et remettre le bit à 0)

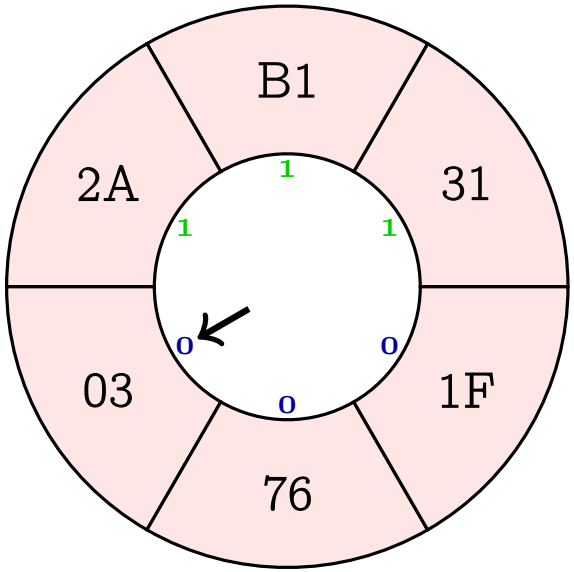
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



Principe

- Bit de 2nd chance mis à 0 lors d'un acces
- Lorsqu'on a besoin de libérer un cadre:
  - Si bit=0, mettre le bit à 1 et passer
  - Sinon utiliser le cadre (et remettre le bit à 0)

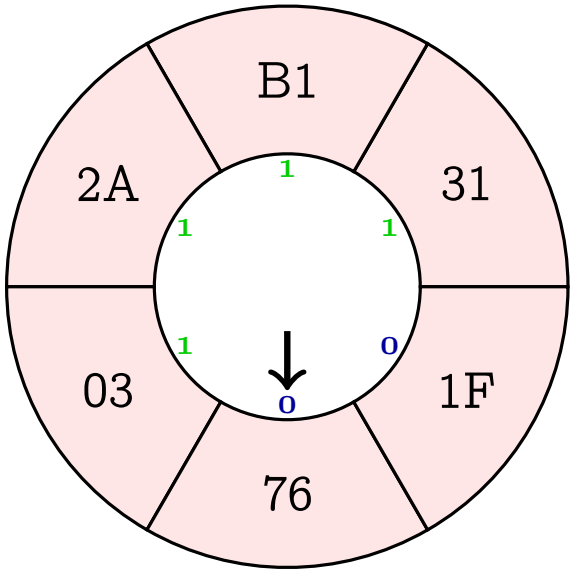
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



Principe

- Bit de 2nd chance mis à 0 lors d'un acces
- Lorsqu'on a besoin de libérer un cadre:
  - Si bit=0, mettre le bit à 1 et passer
  - Sinon utiliser le cadre (et remettre le bit à 0)

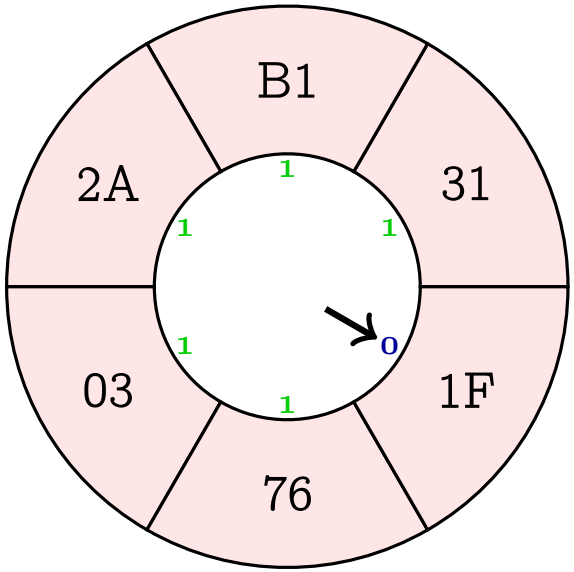
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



Principe

- Bit de 2nd chance mis à 0 lors d'un acces
- Lorsqu'on a besoin de libérer un cadre:
  - Si bit=0, mettre le bit à 1 et passer
  - Sinon utiliser le cadre (et remettre le bit à 0)

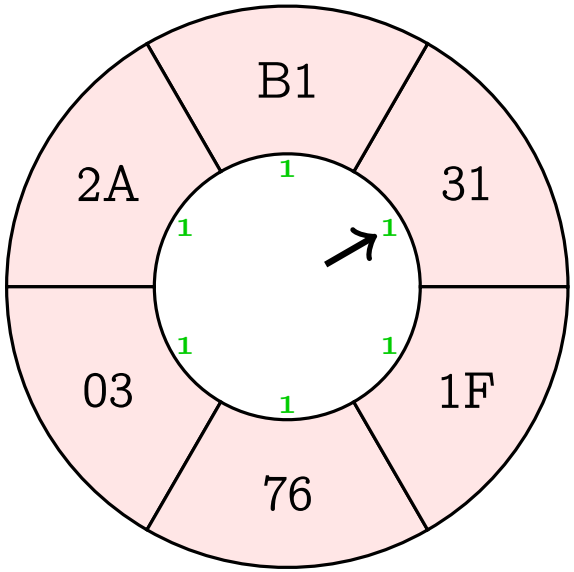
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



Principe

- Bit de 2nd chance mis à 0 lors d'un acces
- Lorsqu'on a besoin de libérer un cadre:
  - Si bit=0, mettre le bit à 1 et passer
  - Sinon utiliser le cadre (et remettre le bit à 0)

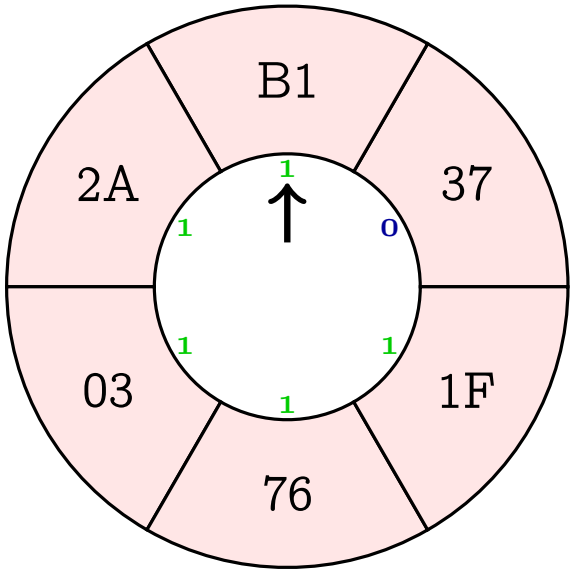
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



Principe

- Bit de 2nd chance mis à 0 lors d'un acces
- Lorsqu'on a besoin de libérer un cadre:
  - Si bit=0, mettre le bit à 1 et passer
  - Sinon utiliser le cadre (et remettre le bit à 0)

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



Algorithmes basés sur l'utilisation

Principe général

Retirer la page la plus pertinente

- La page la moins utilisée (LFU)
- Une page peu utilisée (NRU)
- La page la moins récemment utilisée (LRU)

Principe général

Retirer la page la plus pertinente

- La page la moins utilisée (LFU)
- Une page peu utilisée (NRU)
- La page la moins récemment utilisée (LRU)

Avantage : moins de défaut de page

En pratique, FIFO-2 fait 15 à 20 % plus de défauts que LRU

Principe général

Retirer la page la plus pertinente

- La page la moins utilisée (LFU)
- Une page peu utilisée (NRU)
- La page la moins récemment utilisée (LRU)

Avantage : moins de défaut de page

En pratique, FIFO-2 fait 15 à 20 % plus de défauts que LRU

Inconvénient : algorithmes plus complexes

- Espace mémoire supplémentaire
- Temps de calcul à chaque appel de page

LFU: Least Frequently Used

Principe

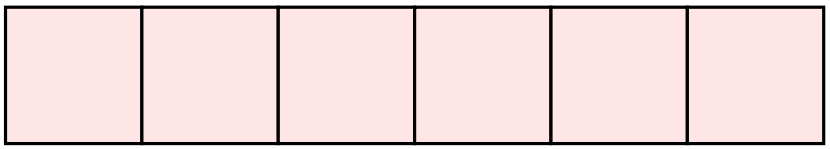
Noter le taux d'utilisation de chaque page du processus.

- choisir la page la moins utilisée

  en cas d'égalité: FIFO

Exemple:

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



Page	Usage	Date
03	0	
04	0	
12	0	
1F	0	
2A	0	
31	0	
37	0	
48	0	
76	0	
B1	0	

LFU: Least Frequently Used

Principe

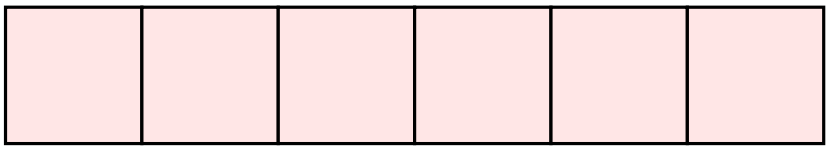
Noter le taux d'utilisation de chaque page du processus.

- choisir la page la moins utilisée

  en cas d'égalité: FIFO

Exemple:

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



Page	Usage	Date
03	0	
04	0	
12	0	
1F	0	
2A	0	
31	0	
37	0	
48	0	
76	0	
B1	0	

LFU: Least Frequently Used

Principe

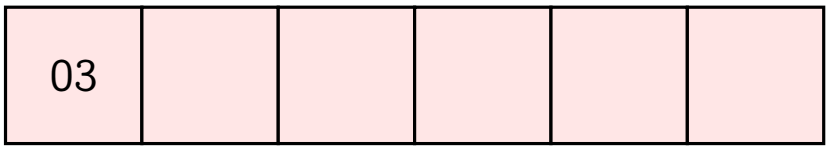
Noter le taux d'utilisation de chaque page du processus.

- choisir la page la moins utilisée

  en cas d'égalité: FIFO

Exemple:

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



Page	Usage	Date
03	1	1
04	0	
12	0	
1F	0	
2A	0	
31	0	
37	0	
48	0	
76	0	
B1	0	

LFU: Least Frequently Used

Principe

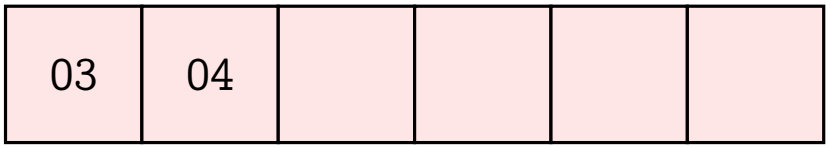
Noter le taux d'utilisation de chaque page du processus.

- choisir la page la moins utilisée

  en cas d'égalité: FIFO

Exemple:

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



Page	Usage	Date
03	1	1
04	1	2
12	0	
1F	0	
2A	0	
31	0	
37	0	
48	0	
76	0	
B1	0	

LFU: Least Frequently Used

Principe

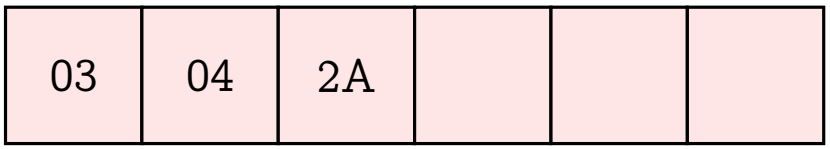
Noter le taux d'utilisation de chaque page du processus.

- choisir la page la moins utilisée

  en cas d'égalité: FIFO

Exemple:

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



Page	Usage	Date
03	1	1
04	1	2
12	0	
1F	0	
2A	1	3
31	0	
37	0	
48	0	
76	0	
B1	0	

LFU: Least Frequently Used

Principe

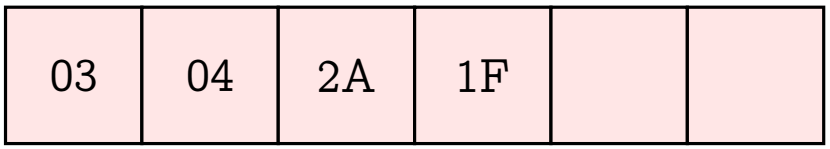
Noter le taux d'utilisation de chaque page du processus.

- choisir la page la moins utilisée

  en cas d'égalité: FIFO

Exemple:

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



Page	Usage	Date
03	1	1
04	1	2
12	0	
1F	1	4
2A	1	3
31	0	
37	0	
48	0	
76	0	
B1	0	

LFU: Least Frequently Used

Principe

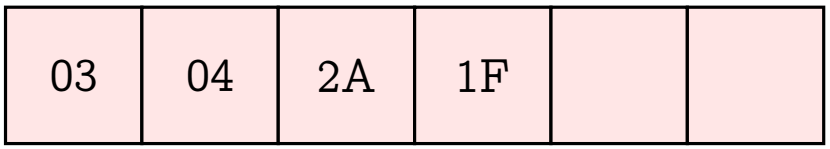
Noter le taux d'utilisation de chaque page du processus.

- choisir la page la moins utilisée

  en cas d'égalité: FIFO

Exemple:

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



Page	Usage	Date
03	1	1
04	1	2
12	0	
1F	1	4
2A	2	3
31	0	
37	0	
48	0	
76	0	
B1	0	

LFU: Least Frequently Used

Principe

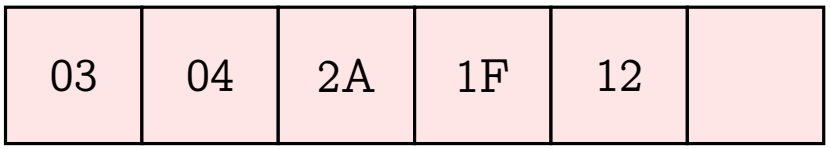
Noter le taux d'utilisation de chaque page du processus.

- choisir la page la moins utilisée

  en cas d'égalité: FIFO

Exemple:

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



Page	Usage	Date
03	1	1
04	1	2
12	1	6
1F	1	4
2A	2	3
31	0	
37	0	
48	0	
76	0	
B1	0	

LFU: Least Frequently Used

Principe

Noter le taux d'utilisation de chaque page du processus.

- choisir la page la moins utilisée

  en cas d'égalité: FIFO

Exemple:

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

03	04	2A	1F	12	48

Page	Usage	Date
03	1	1
04	1	2
12	1	6
1F	1	4
2A	2	3
31	0	
37	0	
48	1	7
76	0	
B1	0	

LFU: Least Frequently Used

Principe

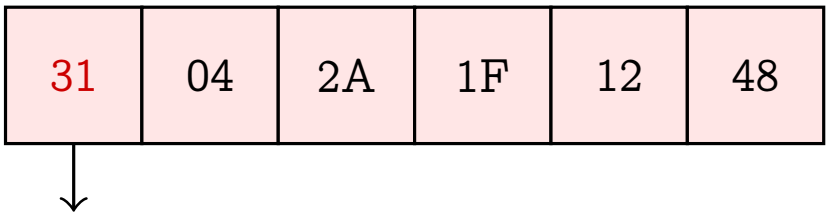
Noter le taux d'utilisation de chaque page du processus.

- choisir la page la moins utilisée

  en cas d'égalité: FIFO

Exemple:

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



Usage = 1, Date = min(1,2,4,6,7)

Page	Usage	Date
03	1	1
04	1	2
12	1	6
1F	1	4
2A	2	3
31	1	8
37	0	
48	1	7
76	0	
B1	0	

LFU: Least Frequently Used

Principe

Noter le taux d'utilisation de chaque page du processus.

- choisir la page la moins utilisée

  en cas d'égalité: FIFO

Exemple:

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

31	B1	2A	1F	12	48

Page	Usage	Date
03	1	1
04	1	2
12	1	6
1F	1	4
2A	2	3
31	1	8
37	0	
48	1	7
76	0	
B1	1	9

LFU: Least Frequently Used

Principe

Noter le taux d'utilisation de chaque page du processus.

- choisir la page la moins utilisée

  en cas d'égalité: FIFO

Exemple:

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

31	B1	2A	1F	12	48

Page	Usage	Date
03	1	1
04	1	2
12	1	6
1F	1	4
2A	3	3
31	1	8
37	0	
48	1	7
76	0	
B1	1	9

LFU: Least Frequently Used

Principe

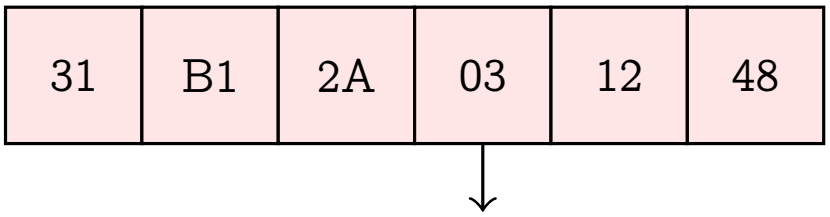
Noter le taux d'utilisation de chaque page du processus.

- choisir la page la moins utilisée

  en cas d'égalité: FIFO

Exemple:

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



Usage = 1, Date = min(8,9,4,6,7)

Page	Usage	Date
03	2	11
04	1	2
12	1	6
1F	1	4
2A	3	3
31	1	8
37	0	
48	1	7
76	0	
B1	1	9

LFU: Least Frequently Used

Principe

Noter le taux d'utilisation de chaque page du processus.

- choisir la page la moins utilisée

  en cas d'égalité: FIFO

Exemple:

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

31	B1	2A	03	76	48

Page	Usage	Date
03	2	11
04	1	2
12	1	6
1F	1	4
2A	3	3
31	1	8
37	0	
48	1	7
76	1	12
B1	1	9

LFU: Least Frequently Used

Principe

Noter le taux d'utilisation de chaque page du processus.

- choisir la page la moins utilisée

  en cas d'égalité: FIFO

Exemple:

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

31	B1	2A	03	76	48

Page	Usage	Date
03	2	11
04	1	2
12	1	6
1F	1	4
2A	4	3
31	1	8
37	0	
48	1	7
76	1	12
B1	1	9

LFU: Least Frequently Used

Principe

Noter le taux d'utilisation de chaque page du processus.

- choisir la page la moins utilisée

  en cas d'égalité: FIFO

Exemple:

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

31	B1	2A	03	76	1F

Page	Usage	Date
03	2	11
04	1	2
12	1	6
1F	2	14
2A	4	3
31	1	8
37	0	
48	1	7
76	1	12
B1	1	9

LFU: Least Frequently Used

Principe

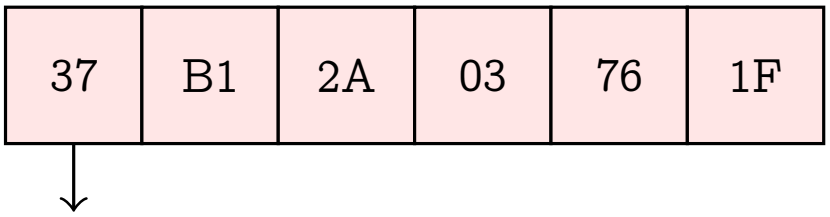
Noter le taux d'utilisation de chaque page du processus.

- choisir la page la moins utilisée

  en cas d'égalité: FIFO

Exemple:

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



Usage = 1, Date = min(8,9,12)

Page	Usage	Date
03	2	11
04	1	2
12	1	6
1F	2	14
2A	4	3
31	1	8
37	1	15
48	1	7
76	1	12
B1	1	9

LFU: avantages et inconvénients

Nombre total d'utilisation

✗ **Problème:** une page beaucoup utilisée reste

✓ **Solution:** remettre usage à 0 périodiquement

fenêtre glissante trop coûteux à implémenter...

LFU: avantages et inconvénients

Nombre total d'utilisation

✗ **Problème:** une page beaucoup utilisée reste

✓ **Solution:** remettre usage à 0 périodiquement

fenêtre glissante trop coûteux à implémenter...

Performances

✓ Beaucoup moins de défaut de page que les autres

LFU: avantages et inconvénients

Nombre total d'utilisation

- ✗ **Problème:** une page beaucoup utilisée reste
- ✓ **Solution:** remettre usage à 0 périodiquement

fenêtre glissante trop coûteux à implémenter...

Performances

- ✓ Beaucoup moins de défaut de page que les autres

Mais aussi

- ✗ 64 bits de plus dans chaque ligne
- ✗ $\mathcal{O}(n)$ à chaque page manquante

NRU: Not Recently Used

Observations

- Compter les utilisations est coûteux

NRU: Not Recently Used

Observations

- Compter les utilisations est coûteux
  $\rightarrow$ *FIFO-2* = 1 seul bit : mémoire + temps!

NRU: Not Recently Used

Observations

- Compter les utilisations est coûteux
  $\rightarrow$ *FIFO-2* = 1 seul bit : mémoire + temps!
- Certaines pages deviennent inutiles

NRU: Not Recently Used

Observations

- Compter les utilisations est coûteux
  $\rightarrow$ *FIFO-2* = 1 seul bit : mémoire + temps!
- Certaines pages deviennent inutiles
  $\rightarrow$ remettre à 0 périodiquement

NRU: Not Recently Used

Observations

- Compter les utilisations est coûteux
  $\rightarrow$ *FIFO-2* = 1 seul bit : mémoire + temps!

- Certaines pages deviennent inutiles
  $\rightarrow$ remettre à 0 périodiquement

- Les pages dans lesquelles on écrit sont plus susceptibles d'êtres réutilisées

un peu plus tard, en général...

NRU: Not Recently Used

Observations

- Compter les utilisations est coûteux
  $\rightarrow$ *FIFO-2* = 1 seul bit : mémoire + temps!

- Certaines pages deviennent inutiles
  $\rightarrow$ remettre à 0 périodiquement

- Les pages dans lesquelles on écrit sont plus susceptibles d'êtres réutilisées

un peu plus tard, en général...

$\rightarrow$ Ajouter un 2e bit pour les pages modifiées

Not Recently Used

2 bits: R (page référencée) et M (page modifiée)

- Lecture ou écriture $\rightarrow$ R=1 ; écriture $\rightarrow$ M=1
- Priorité: *(FIFO en cas d'égalité)*

  (R=1,M=1) > (R=1,M=0) > (R=0,M=1) > (R=0,M=0)
- Tous les R sont remis à 0 chaque K cycles

03r 04w 2Ar 1Fw 2Aw 12r 48r 31r B1r 2Ar 03w 76r 2Aw 1Fw 37r

Page	R	M	Date
03	0	0	
04	0	0	
12	0	0	
1F	0	0	
2A	0	0	
31	0	0	
37	0	0	
48	0	0	
76	0	0	
B1	0	0	

Not Recently Used

2 bits: R (page référencée) et M (page modifiée)

- Lecture ou écriture → R=1 ; écriture → M=1
- Priorité: *(FIFO en cas d'égalité)*

  (R=1,M=1) > (R=1,M=0) > (R=0,M=1) > (R=0,M=0)
- Tous les R sont remis à 0 chaque K cycles

03r 04w 2Ar 1Fw 2Aw 12r 48r 31r B1r 2Ar 03w 76r 2Aw 1Fw 37r

Page	R	M	Date
03	0	0	
04	0	0	
12	0	0	
1F	0	0	
2A	0	0	
31	0	0	
37	0	0	
48	0	0	
76	0	0	
B1	0	0	

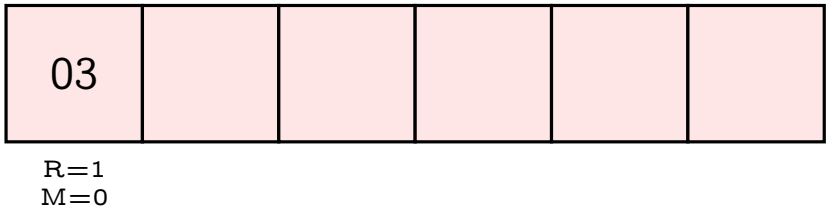
Not Recently Used

2 bits: R (page référencée) et M (page modifiée)

- Lecture ou écriture → R=1 ; écriture → M=1
- Priorité: *(FIFO en cas d'égalité)*

  (R=1,M=1) > (R=1,M=0) > (R=0,M=1) > (R=0,M=0)
- Tous les R sont remis à 0 chaque K cycles

03r 04w 2Ar 1Fw 2Aw 12r 48r 31r B1r 2Ar 03w 76r 2Aw 1Fw 37r



Page	R	M	Date
03	1	0	1
04	0	0	
12	0	0	
1F	0	0	
2A	0	0	
31	0	0	
37	0	0	
48	0	0	
76	0	0	
B1	0	0	

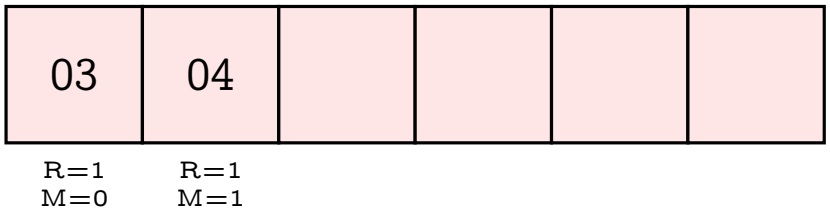
Not Recently Used

2 bits: R (page référencée) et M (page modifiée)

- Lecture ou écriture → R=1 ; écriture → M=1
- Priorité: *(FIFO en cas d'égalité)*

  (R=1,M=1) > (R=1,M=0) > (R=0,M=1) > (R=0,M=0)
- Tous les R sont remis à 0 chaque K cycles

03r 04w 2Ar 1Fw 2Aw 12r 48r 31r B1r 2Ar 03w 76r 2Aw 1Fw 37r



Page	R	M	Date
03	1	0	1
04	1	1	2
12	0	0	
1F	0	0	
2A	0	0	
31	0	0	
37	0	0	
48	0	0	
76	0	0	
B1	0	0	

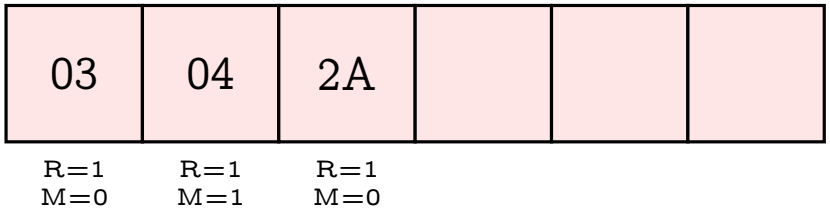
Not Recently Used

2 bits: R (page référencée) et M (page modifiée)

- Lecture ou écriture → R=1 ; écriture → M=1
- Priorité: *(FIFO en cas d'égalité)*

  (R=1,M=1) > (R=1,M=0) > (R=0,M=1) > (R=0,M=0)
- Tous les R sont remis à 0 chaque K cycles

03r 04w 2Ar 1Fw 2Aw 12r 48r 31r B1r 2Ar 03w 76r 2Aw 1Fw 37r



Page	R	M	Date
03	1	0	1
04	1	1	2
12	0	0	
1F	0	0	
2A	1	0	3
31	0	0	
37	0	0	
48	0	0	
76	0	0	
B1	0	0	

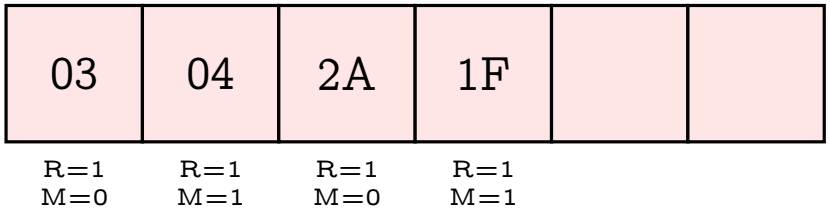
Not Recently Used

2 bits: R (page référencée) et M (page modifiée)

- Lecture ou écriture → R=1 ; écriture → M=1
- Priorité: *(FIFO en cas d'égalité)*

  (R=1,M=1) > (R=1,M=0) > (R=0,M=1) > (R=0,M=0)
- Tous les R sont remis à 0 chaque K cycles

03r 04w 2Ar 1Fw 2Aw 12r 48r 31r B1r 2Ar 03w 76r 2Aw 1Fw 37r



Page	R	M	Date
03	1	0	1
04	1	1	2
12	0	0	
1F	1	1	4
2A	1	0	3
31	0	0	
37	0	0	
48	0	0	
76	0	0	
B1	0	0	

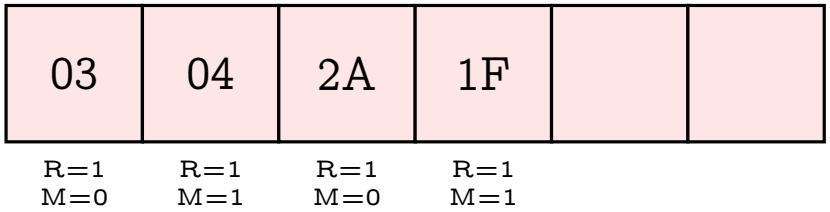
Not Recently Used

2 bits: R (page référencée) et M (page modifiée)

- Lecture ou écriture → R=1 ; écriture → M=1
- Priorité: *(FIFO en cas d'égalité)*

  (R=1,M=1) > (R=1,M=0) > (R=0,M=1) > (R=0,M=0)
- Tous les R sont remis à 0 chaque K cycles

03r 04w 2Ar 1Fw 2Aw 12r 48r 31r B1r 2Ar 03w 76r 2Aw 1Fw 37r



Page	R	M	Date
03	1	0	1
04	1	1	2
12	0	0	
1F	1	1	4
2A	1	0	3
31	0	0	
37	0	0	
48	0	0	
76	0	0	
B1	0	0	

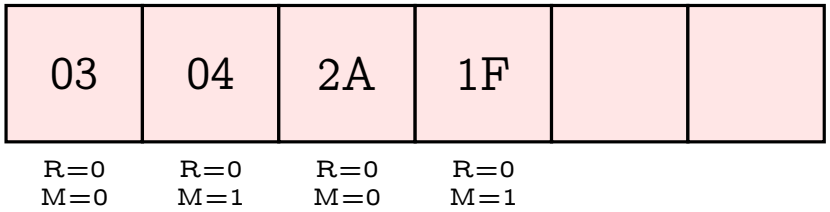
Not Recently Used

2 bits: R (page référencée) et M (page modifiée)

- Lecture ou écriture → R=1 ; écriture → M=1
- Priorité: *(FIFO en cas d'égalité)*

  (R=1,M=1) > (R=1,M=0) > (R=0,M=1) > (R=0,M=0)
- Tous les R sont remis à 0 chaque K cycles

03r 04w 2Ar 1Fw 2Aw 12r 48r 31r B1r 2Ar 03w 76r 2Aw 1Fw 37r



RESET

Page	R	M	Date
03	0	0	1
04	0	1	2
12	0	0	
1F	0	1	4
2A	0	0	3
31	0	0	
37	0	0	
48	0	0	
76	0	0	
B1	0	0	

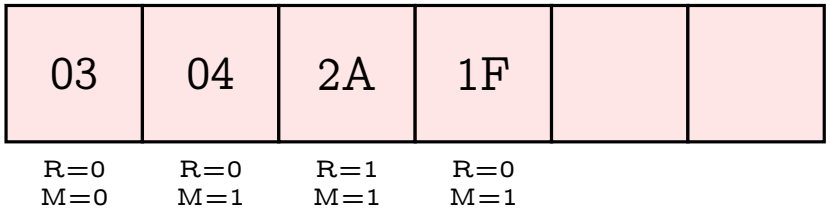
Not Recently Used

2 bits: R (page référencée) et M (page modifiée)

- Lecture ou écriture → R=1 ; écriture → M=1
- Priorité: *(FIFO en cas d'égalité)*

  (R=1,M=1) > (R=1,M=0) > (R=0,M=1) > (R=0,M=0)
- Tous les R sont remis à 0 chaque K cycles

03r 04w 2Ar 1Fw 2Aw 12r 48r 31r B1r 2Ar 03w 76r 2Aw 1Fw 37r



Page	R	M	Date
03	0	0	1
04	0	1	2
12	0	0	
1F	0	1	4
2A	1	1	3
31	0	0	
37	0	0	
48	0	0	
76	0	0	
B1	0	0	

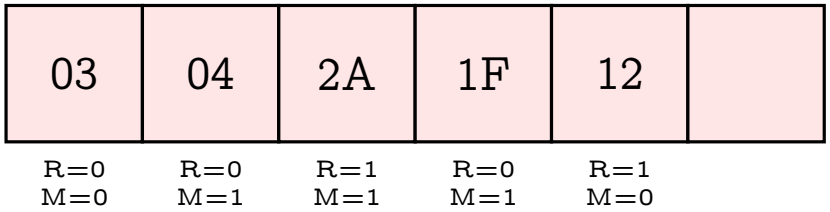
Not Recently Used

2 bits: R (page référencée) et M (page modifiée)

- Lecture ou écriture → R=1 ; écriture → M=1
- Priorité: *(FIFO en cas d'égalité)*

  (R=1,M=1) > (R=1,M=0) > (R=0,M=1) > (R=0,M=0)
- Tous les R sont remis à 0 chaque K cycles

03r 04w 2Ar 1Fw 2Aw 12r 48r 31r B1r 2Ar 03w 76r 2Aw 1Fw 37r



Page	R	M	Date
03	0	0	1
04	0	1	2
12	1	0	6
1F	0	1	4
2A	1	1	3
31	0	0	
37	0	0	
48	0	0	
76	0	0	
B1	0	0	

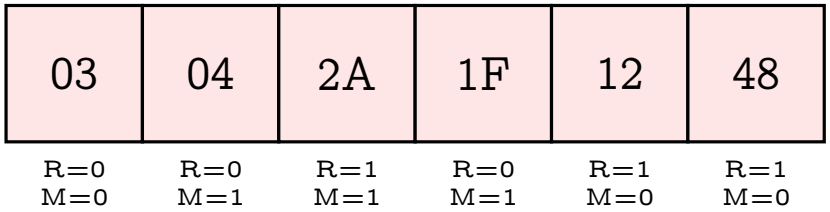
Not Recently Used

2 bits: R (page référencée) et M (page modifiée)

- Lecture ou écriture → R=1 ; écriture → M=1
- Priorité: *(FIFO en cas d'égalité)*

  (R=1,M=1) > (R=1,M=0) > (R=0,M=1) > (R=0,M=0)
- Tous les R sont remis à 0 chaque K cycles

03r 04w 2Ar 1Fw 2Aw 12r 48r 31r B1r 2Ar 03w 76r 2Aw 1Fw 37r



Page	R	M	Date
03	0	0	1
04	0	1	2
12	1	0	6
1F	0	1	4
2A	1	1	3
31	0	0	
37	0	0	
48	1	0	7
76	0	0	
B1	0	0	

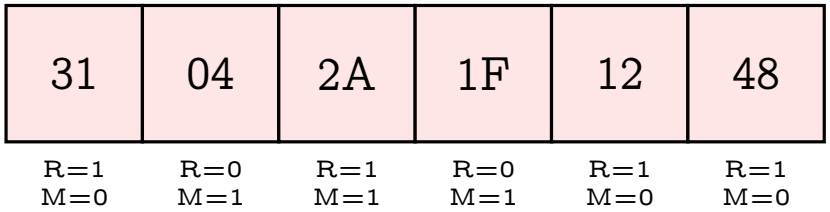
Not Recently Used

2 bits: R (page référencée) et M (page modifiée)

- Lecture ou écriture → R=1 ; écriture → M=1
- Priorité: *(FIFO en cas d'égalité)*

  (R=1,M=1) > (R=1,M=0) > (R=0,M=1) > (R=0,M=0)
- Tous les R sont remis à 0 chaque K cycles

03r 04w 2Ar 1Fw 2Aw 12r 48r 31r B1r 2Ar 03w 76r 2Aw 1Fw 37r



Page	R	M	Date
03	0	0	1
04	0	1	2
12	1	0	6
1F	0	1	4
2A	1	1	3
31	1	0	8
37	0	0	
48	1	0	7
76	0	0	
B1	0	0	

Not Recently Used

2 bits: R (page référencée) et M (page modifiée)

- Lecture ou écriture → R=1 ; écriture → M=1
- Priorité: *(FIFO en cas d'égalité)*

  (R=1,M=1) > (R=1,M=0) > (R=0,M=1) > (R=0,M=0)
- Tous les R sont remis à 0 chaque K cycles

03r 04w 2Ar 1Fw 2Aw 12r 48r 31r B1r 2Ar 03w 76r 2Aw 1Fw 37r

31	04	2A	1F	12	48
R=1 M=0	R=0 M=1	R=1 M=1	R=0 M=1	R=1 M=0	R=1 M=0

RESET

Page	R	M	Date
03	0	0	1
04	0	1	2
12	1	0	6
1F	0	1	4
2A	1	1	3
31	1	0	8
37	0	0	
48	1	0	7
76	0	0	
B1	0	0	

Not Recently Used

2 bits: R (page référencée) et M (page modifiée)

- Lecture ou écriture → R=1 ; écriture → M=1
- Priorité: *(FIFO en cas d'égalité)*

  (R=1,M=1) > (R=1,M=0) > (R=0,M=1) > (R=0,M=0)
- Tous les R sont remis à 0 chaque K cycles

03r 04w 2Ar 1Fw 2Aw 12r 48r 31r B1r 2Ar 03w 76r 2Aw 1Fw 37r

31	04	2A	1F	12	48
R=0 M=0	R=0 M=1	R=0 M=1	R=0 M=1	R=0 M=0	R=0 M=0

RESET

Page	R	M	Date
03	0	0	1
04	0	1	2
12	0	0	6
1F	0	1	4
2A	0	1	3
31	0	0	8
37	0	0	
48	0	0	7
76	0	0	
B1	0	0	

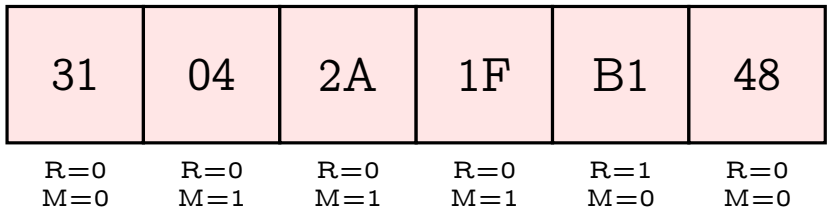
Not Recently Used

2 bits: R (page référencée) et M (page modifiée)

- Lecture ou écriture → R=1 ; écriture → M=1
- Priorité: *(FIFO en cas d'égalité)*

  (R=1,M=1) > (R=1,M=0) > (R=0,M=1) > (R=0,M=0)
- Tous les R sont remis à 0 chaque K cycles

03r 04w 2Ar 1Fw 2Aw 12r 48r 31r B1r 2Ar 03w 76r 2Aw 1Fw 37r



Page	R	M	Date
03	0	0	1
04	0	1	2
12	0	0	6
1F	0	1	4
2A	0	1	3
31	0	0	8
37	0	0	
48	0	0	7
76	0	0	
B1	1	0	9

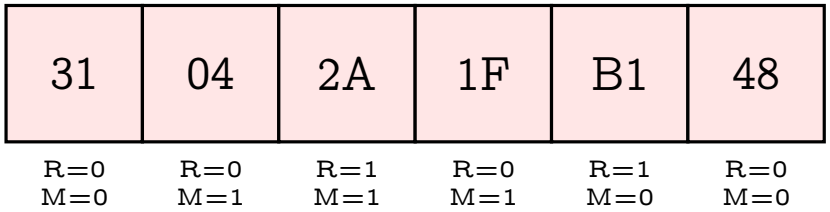
Not Recently Used

2 bits: R (page référencée) et M (page modifiée)

- Lecture ou écriture → R=1 ; écriture → M=1
- Priorité: *(FIFO en cas d'égalité)*

  (R=1,M=1) > (R=1,M=0) > (R=0,M=1) > (R=0,M=0)
- Tous les R sont remis à 0 chaque K cycles

03r 04w 2Ar 1Fw 2Aw 12r 48r 31r B1r 2Ar 03w 76r 2Aw 1Fw 37r



Page	R	M	Date
03	0	0	1
04	0	1	2
12	0	0	6
1F	0	1	4
2A	1	1	3
31	0	0	8
37	0	0	
48	0	0	7
76	0	0	
B1	1	0	9

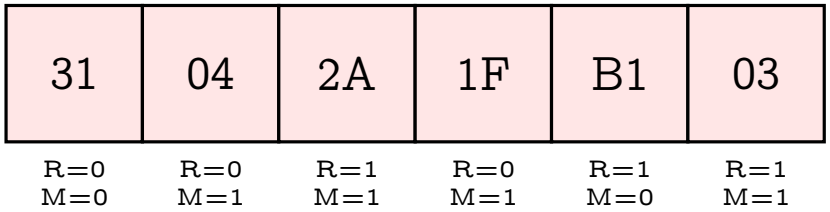
Not Recently Used

2 bits: R (page référencée) et M (page modifiée)

- Lecture ou écriture → R=1 ; écriture → M=1
- Priorité: *(FIFO en cas d'égalité)*

  (R=1,M=1) > (R=1,M=0) > (R=0,M=1) > (R=0,M=0)
- Tous les R sont remis à 0 chaque K cycles

03r 04w 2Ar 1Fw 2Aw 12r 48r 31r B1r 2Ar 03w 76r 2Aw 1Fw 37r



Page	R	M	Date
03	1	1	11
04	0	1	2
12	0	0	6
1F	0	1	4
2A	1	1	3
31	0	0	8
37	0	0	
48	0	0	7
76	0	0	
B1	1	0	9

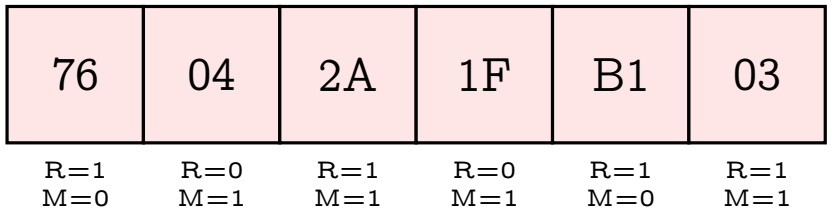
Not Recently Used

2 bits: R (page référencée) et M (page modifiée)

- Lecture ou écriture → R=1 ; écriture → M=1
- Priorité: *(FIFO en cas d'égalité)*

  (R=1,M=1) > (R=1,M=0) > (R=0,M=1) > (R=0,M=0)
- Tous les R sont remis à 0 chaque K cycles

03r 04w 2Ar 1Fw 2Aw 12r 48r 31r B1r 2Ar 03w 76r 2Aw 1Fw 37r



Page	R	M	Date
03	1	1	11
04	0	1	2
12	0	0	6
1F	0	1	4
2A	1	1	3
31	0	0	8
37	0	0	
48	0	0	7
76	1	0	12
B1	1	0	9

Not Recently Used

2 bits: R (page référencée) et M (page modifiée)

- Lecture ou écriture → R=1 ; écriture → M=1
- Priorité: *(FIFO en cas d'égalité)*

  (R=1,M=1) > (R=1,M=0) > (R=0,M=1) > (R=0,M=0)
- Tous les R sont remis à 0 chaque K cycles

03r 04w 2Ar 1Fw 2Aw 12r 48r 31r B1r 2Ar 03w 76r 2Aw 1Fw 37r

76	04	2A	1F	B1	03
R=1 M=0	R=0 M=1	R=1 M=1	R=0 M=1	R=1 M=0	R=1 M=1

RESET

Page	R	M	Date
03	1	1	11
04	0	1	2
12	0	0	6
1F	0	1	4
2A	1	1	3
31	0	0	8
37	0	0	
48	0	0	7
76	1	0	12
B1	1	0	9

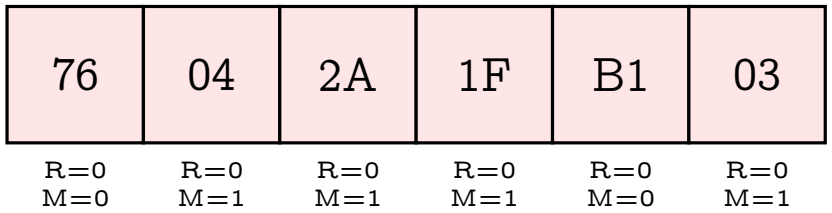
Not Recently Used

2 bits: R (page référencée) et M (page modifiée)

- Lecture ou écriture → R=1 ; écriture → M=1
- Priorité: *(FIFO en cas d'égalité)*

  (R=1,M=1) > (R=1,M=0) > (R=0,M=1) > (R=0,M=0)
- Tous les R sont remis à 0 chaque K cycles

03r 04w 2Ar 1Fw 2Aw 12r 48r 31r B1r 2Ar 03w 76r 2Aw 1Fw 37r



RESET

Page	R	M	Date
03	0	1	11
04	0	1	2
12	0	0	6
1F	0	1	4
2A	0	1	3
31	0	0	8
37	0	0	
48	0	0	7
76	0	0	12
B1	0	0	9

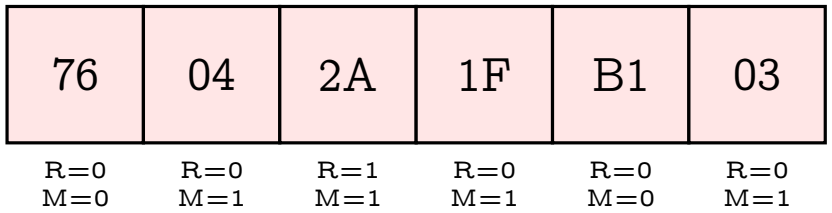
Not Recently Used

2 bits: R (page référencée) et M (page modifiée)

- Lecture ou écriture → R=1 ; écriture → M=1
- Priorité: *(FIFO en cas d'égalité)*

  (R=1,M=1) > (R=1,M=0) > (R=0,M=1) > (R=0,M=0)
- Tous les R sont remis à 0 chaque K cycles

03r 04w 2Ar 1Fw 2Aw 12r 48r 31r B1r 2Ar 03w 76r 2Aw 1Fw 37r



Page	R	M	Date
03	0	1	11
04	0	1	2
12	0	0	6
1F	0	1	4
2A	1	1	3
31	0	0	8
37	0	0	
48	0	0	7
76	0	0	12
B1	0	0	9

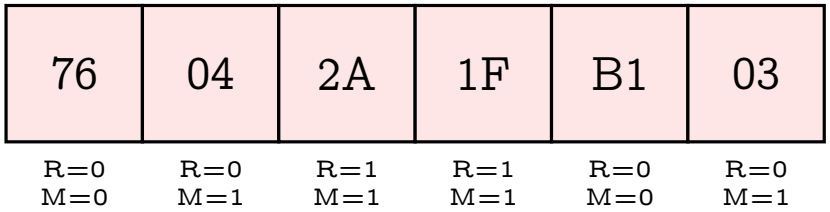
Not Recently Used

2 bits: R (page référencée) et M (page modifiée)

- Lecture ou écriture → R=1 ; écriture → M=1
- Priorité: *(FIFO en cas d'égalité)*

  (R=1,M=1) > (R=1,M=0) > (R=0,M=1) > (R=0,M=0)
- Tous les R sont remis à 0 chaque K cycles

03r 04w 2Ar 1Fw 2Aw 12r 48r 31r B1r 2Ar 03w 76r 2Aw 1Fw 37r



Page	R	M	Date
03	0	1	11
04	0	1	2
12	0	0	6
1F	1	1	4
2A	1	1	3
31	0	0	8
37	0	0	
48	0	0	7
76	0	0	12
B1	0	0	9

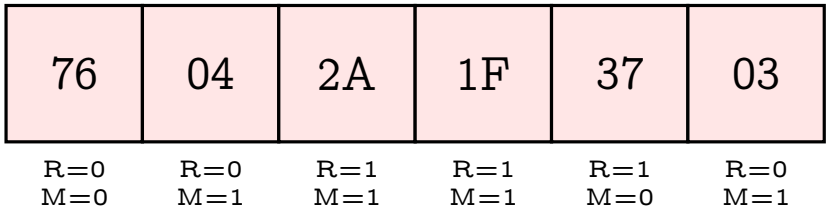
Not Recently Used

2 bits: R (page référencée) et M (page modifiée)

- Lecture ou écriture → R=1 ; écriture → M=1
- Priorité: *(FIFO en cas d'égalité)*

  (R=1,M=1) > (R=1,M=0) > (R=0,M=1) > (R=0,M=0)
- Tous les R sont remis à 0 chaque K cycles

03r 04w 2Ar 1Fw 2Aw 12r 48r 31r B1r 2Ar 03w 76r 2Aw 1Fw 37r



Page	R	M	Date
03	0	1	11
04	0	1	2
12	0	0	6
1F	1	1	4
2A	1	1	3
31	0	0	8
37	1	0	15
48	0	0	7
76	0	0	12
B1	0	0	9

NRU: avantages et inconvénients

Performance

✓ Peu de défaut de page

✓ Peu coûteux en mémoire

X Temps de calcul: $\mathcal{O}(n)$ à chaque reset et page fault

NRU: avantages et inconvénients

Performance

✓ Peu de défaut de page

✓ Peu coûteux en mémoire

X Temps de calcul: $\mathcal{O}(n)$ à chaque reset et page fault

Gain

En pratique, gain trop faible par rapport à FIFO-2

LRU: Least Recently Used

Principe

- File (FIFO) avec remise en fin
- Implémentation matérielle $\rightarrow$ calcul en $\mathcal{O}(1)$

LRU: Least Recently Used

Principe

- File (FIFO) avec remise en fin
- Implémentation matérielle $\rightarrow$ calcul en $\mathcal{O}(1)$

Implémentation

- Matrice triangulaire N = nombre de cadres

  sans la diagonale, tout initialisé à 0

	0	1	2	3	4	5
0		0	0	0	0	0
1			0	0	0	0
2				0	0	0
3					0	0
4						0
5						

Implémentation

- Matrice triangulaire N = nombre de cadres
- Utilisation d'une page dans le cadre i
  $\rightarrow$ ligne i à 1 puis colonne i à 0
- Cadre le plus ancien (à utiliser) =
  ligne est remplie de 0, colonne remplie de 1 (sauf ligne i)

	0	1	2	3	4	5
0		0	0	0	0	0
1			0	0	0	0
2				0	0	0
3					0	0
4						0
5						

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

Implémentation

- Matrice triangulaire N = nombre de cadres
- Utilisation d'une page dans le cadre i
  $\rightarrow$ ligne i à 1 puis colonne i à 0
- Cadre le plus ancien (à utiliser) =
  ligne est remplie de 0, colonne remplie de 1 (sauf ligne i)

	0	1	2	3	4	5
0		0	0	0	0	0
1			0	0	0	0
2				0	0	0
3					0	0
4						0
5						

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

Implémentation

- Matrice triangulaire N = nombre de cadres
- Utilisation d'une page dans le cadre i
  $\rightarrow$ ligne i à 1 puis colonne i à 0
- Cadre le plus ancien (à utiliser) =
  ligne est remplie de 0, colonne remplie de 1 (sauf ligne i)

	0	1	2	3	4	5
0		1	1	1	1	1
1			0	0	0	0
2				0	0	0
3					0	0
4						0
5						

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

Implémentation

- Matrice triangulaire N = nombre de cadres
- Utilisation d'une page dans le cadre i
  $\rightarrow$ ligne i à 1 puis colonne i à 0
- Cadre le plus ancien (à utiliser) =
  ligne est remplie de 0, colonne remplie de 1 (sauf ligne i)

	0	1	2	3	4	5
0		1	1	1	1	1
1			0	0	0	0
2				0	0	0
3					0	0
4						0
5						

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

Implémentation

- Matrice triangulaire N = nombre de cadres
- Utilisation d'une page dans le cadre i
  $\rightarrow$ ligne i à 1 puis colonne i à 0
- Cadre le plus ancien (à utiliser) =
  ligne est remplie de 0, colonne remplie de 1 (sauf ligne i)

	0	1	2	3	4	5
0		1	1	1	1	1
1			0	0	0	0
2				0	0	0
3					0	0
4						0
5						

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

Implémentation

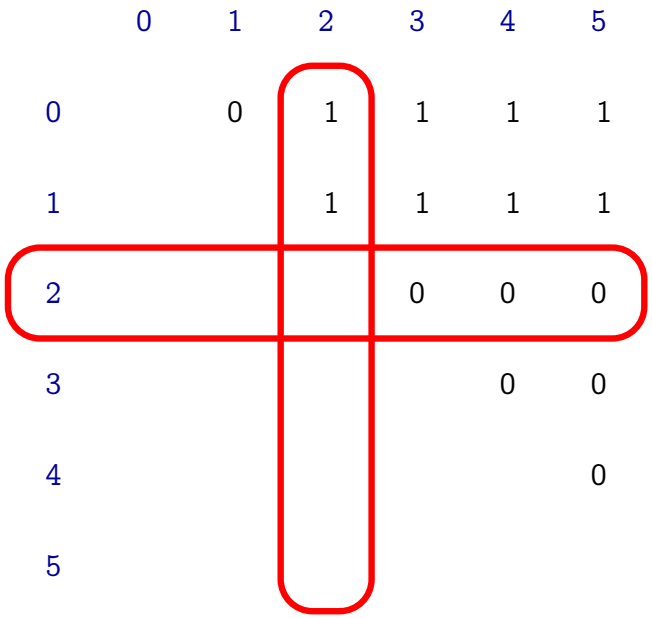
- Matrice triangulaire N = nombre de cadres
- Utilisation d'une page dans le cadre i
  $\rightarrow$ ligne i à 1 puis colonne i à 0
- Cadre le plus ancien (à utiliser) =
  ligne est remplie de 0, colonne remplie de 1 (sauf ligne i)

	0	1	2	3	4	5
0		0	1	1	1	1
1			1	1	1	1
2				0	0	0
3					0	0
4						0
5						

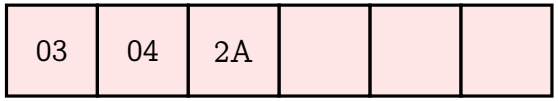
03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

Implémentation

- Matrice triangulaire N = nombre de cadres
- Utilisation d'une page dans le cadre i
  $\rightarrow$ ligne i à 1 puis colonne i à 0
- Cadre le plus ancien (à utiliser) =
  ligne est remplie de 0, colonne remplie de 1 (sauf ligne i)



03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

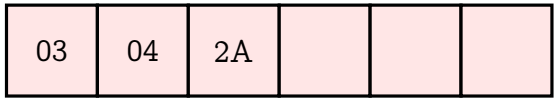


Implémentation

- Matrice triangulaire N = nombre de cadres
- Utilisation d'une page dans le cadre i
  $\rightarrow$ ligne i à 1 puis colonne i à 0
- Cadre le plus ancien (à utiliser) =
  ligne est remplie de 0, colonne remplie de 1 (sauf ligne i)

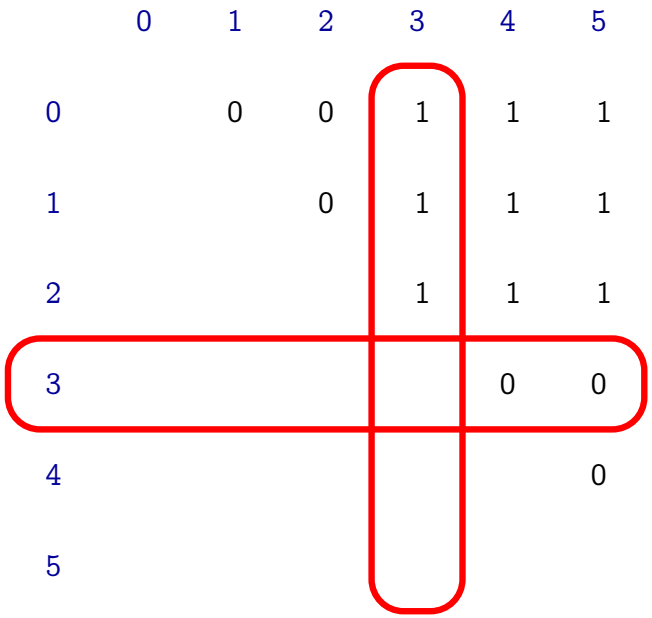
	0	1	2	3	4	5
0		0	0	1	1	1
1			0	1	1	1
2				1	1	1
3					0	0
4						0
5						

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

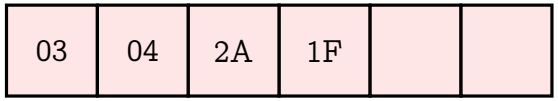


Implémentation

- Matrice triangulaire N = nombre de cadres
- Utilisation d'une page dans le cadre i
  $\rightarrow$ ligne i à 1 puis colonne i à 0
- Cadre le plus ancien (à utiliser) =
  ligne est remplie de 0, colonne remplie de 1 (sauf ligne i)



03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

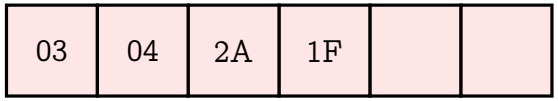


Implémentation

- Matrice triangulaire N = nombre de cadres
- Utilisation d'une page dans le cadre i
  $\rightarrow$ ligne i à 1 puis colonne i à 0
- Cadre le plus ancien (à utiliser) =
  ligne est remplie de 0, colonne remplie de 1 (sauf ligne i)

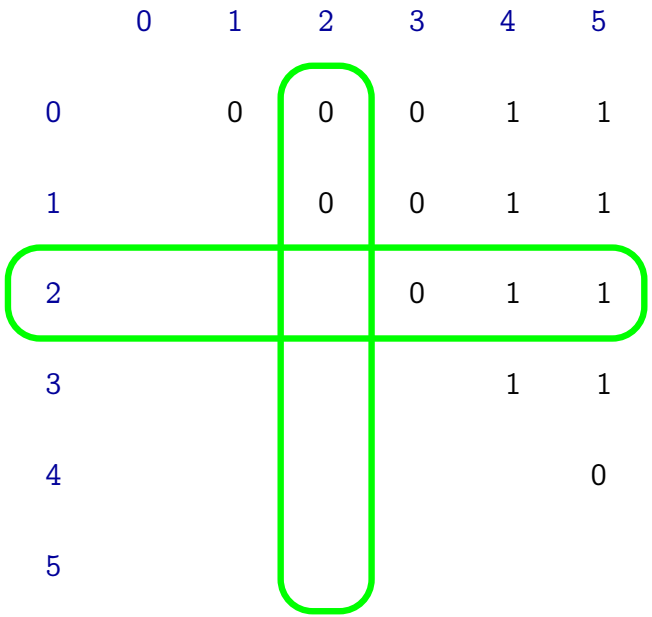
	0	1	2	3	4	5
0		0	0	0	1	1
1			0	0	1	1
2				0	1	1
3					1	1
4						0
5						

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

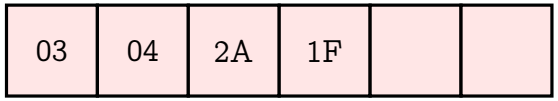


Implémentation

- Matrice triangulaire N = nombre de cadres
- Utilisation d'une page dans le cadre i
  $\rightarrow$ ligne i à 1 puis colonne i à 0
- Cadre le plus ancien (à utiliser) =
  ligne est remplie de 0, colonne remplie de 1 (sauf ligne i)



03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



Implémentation

- Matrice triangulaire N = nombre de cadres
- Utilisation d'une page dans le cadre i
  $\rightarrow$ ligne i à 1 puis colonne i à 0
- Cadre le plus ancien (à utiliser) =
  ligne est remplie de 0, colonne remplie de 1 (sauf ligne i)

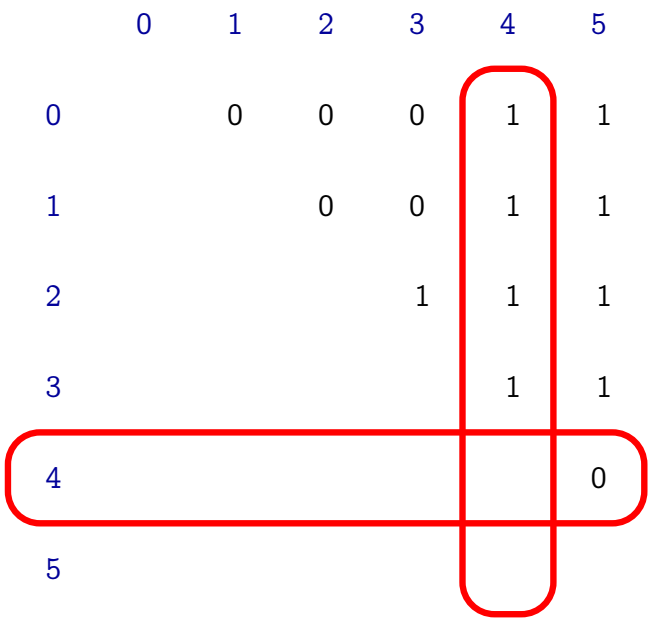
	0	1	2	3	4	5
0		0	0	0	1	1
1			0	0	1	1
2				1	1	1
3					1	1
4						0
5						

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

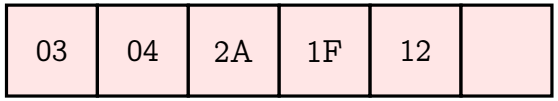
03	04	2A	1F		

Implémentation

- Matrice triangulaire N = nombre de cadres
- Utilisation d'une page dans le cadre i
  $\rightarrow$ ligne i à 1 puis colonne i à 0
- Cadre le plus ancien (à utiliser) =
  ligne est remplie de 0, colonne remplie de 1 (sauf ligne i)



03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



Implémentation

- Matrice triangulaire N = nombre de cadres
- Utilisation d'une page dans le cadre i
  $\rightarrow$ ligne i à 1 puis colonne i à 0
- Cadre le plus ancien (à utiliser) =
  ligne est remplie de 0, colonne remplie de 1 (sauf ligne i)

	0	1	2	3	4	5
0		0	0	0	0	1
1			0	0	0	1
2				1	0	1
3					0	1
4						1
5						

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

03	04	2A	1F	12	

Implémentation

- Matrice triangulaire N = nombre de cadres
- Utilisation d'une page dans le cadre i
  $\rightarrow$ ligne i à 1 puis colonne i à 0
- Cadre le plus ancien (à utiliser) =
  ligne est remplie de 0, colonne remplie de 1 (sauf ligne i)

	0	1	2	3	4	5
0		0	0	0	0	1
1			0	0	0	1
2				1	0	1
3					0	1
4						1
5						

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

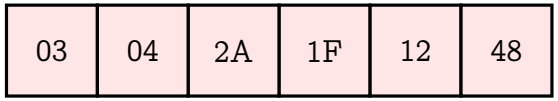
03	04	2A	1F	12	48

Implémentation

- Matrice triangulaire N = nombre de cadres
- Utilisation d'une page dans le cadre i
  $\rightarrow$ ligne i à 1 puis colonne i à 0
- Cadre le plus ancien (à utiliser) =
  ligne est remplie de 0, colonne remplie de 1 (sauf ligne i)

	0	1	2	3	4	5
0		0	0	0	0	0
1			0	0	0	0
2				1	0	0
3					0	0
4						0
5						

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37



Implémentation

- Matrice triangulaire N = nombre de cadres
- Utilisation d'une page dans le cadre i
  $\rightarrow$ ligne i à 1 puis colonne i à 0
- Cadre le plus ancien (à utiliser) =
  ligne est remplie de 0, colonne remplie de 1 (sauf ligne i)

	0	1	2	3	4	5
0		0	0	0	0	0
1			0	0	0	0
2				1	0	0
3					0	0
4						0
5						

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

31	04	2A	1F	12	48

Implémentation

- Matrice triangulaire N = nombre de cadres
- Utilisation d'une page dans le cadre i
  $\rightarrow$ ligne i à 1 puis colonne i à 0
- Cadre le plus ancien (à utiliser) =
  ligne est remplie de 0, colonne remplie de 1 (sauf ligne i)

	0	1	2	3	4	5
0		1	1	1	1	1
1			0	0	0	0
2				1	0	0
3					0	0
4						0
5						

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

31	04	2A	1F	12	48

Implémentation

- Matrice triangulaire N = nombre de cadres
- Utilisation d'une page dans le cadre i
  $\rightarrow$ ligne i à 1 puis colonne i à 0
- Cadre le plus ancien (à utiliser) =
  ligne est remplie de 0, colonne remplie de 1 (sauf ligne i)

	0	1	2	3	4	5
0		1	1	1	1	1
1			0	0	0	0
2				1	0	0
3					0	0
4						0
5						

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

31	B1	2A	1F	12	48

Implémentation

- Matrice triangulaire N = nombre de cadres
- Utilisation d'une page dans le cadre i
  $\rightarrow$ ligne i à 1 puis colonne i à 0
- Cadre le plus ancien (à utiliser) =
  ligne est remplie de 0, colonne remplie de 1 (sauf ligne i)

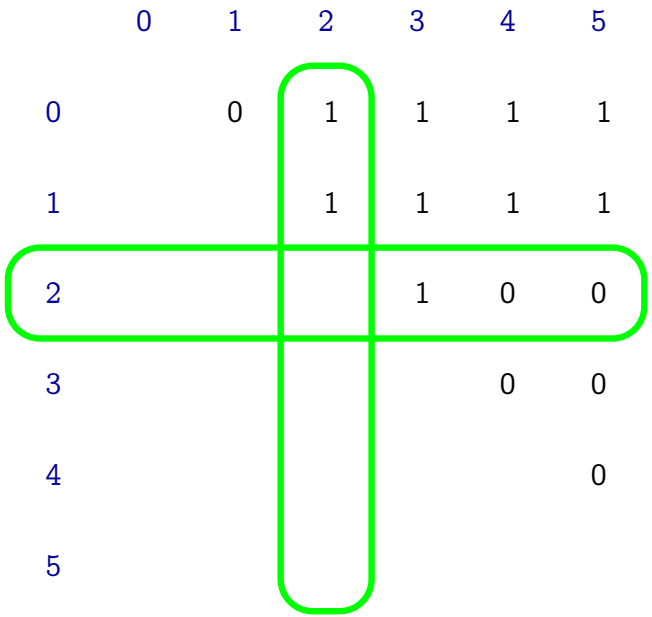
	0	1	2	3	4	5
0		0	1	1	1	1
1			1	1	1	1
2				1	0	0
3					0	0
4						0
5						

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

31	B1	2A	1F	12	48

Implémentation

- Matrice triangulaire N = nombre de cadres
- Utilisation d'une page dans le cadre i
  $\rightarrow$ ligne i à 1 puis colonne i à 0
- Cadre le plus ancien (à utiliser) =
  ligne est remplie de 0, colonne remplie de 1 (sauf ligne i)



03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

31	B1	2A	1F	12	48

Implémentation

- Matrice triangulaire N = nombre de cadres
- Utilisation d'une page dans le cadre i
  $\rightarrow$ ligne i à 1 puis colonne i à 0
- Cadre le plus ancien (à utiliser) =
  ligne est remplie de 0, colonne remplie de 1 (sauf ligne i)

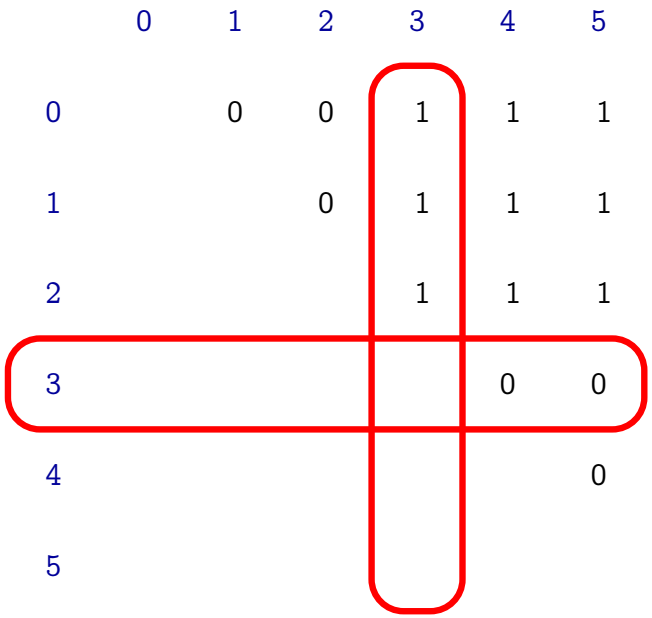
	0	1	2	3	4	5
0		0	0	1	1	1
1			0	1	1	1
2				1	1	1
3					0	0
4						0
5						

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

31	B1	2A	1F	12	48

Implémentation

- Matrice triangulaire N = nombre de cadres
- Utilisation d'une page dans le cadre i
  $\rightarrow$ ligne i à 1 puis colonne i à 0
- Cadre le plus ancien (à utiliser) =
  ligne est remplie de 0, colonne remplie de 1 (sauf ligne i)



03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

31	B1	2A	03	12	48

Implémentation

- Matrice triangulaire N = nombre de cadres
- Utilisation d'une page dans le cadre i
  $\rightarrow$ ligne i à 1 puis colonne i à 0
- Cadre le plus ancien (à utiliser) =
  ligne est remplie de 0, colonne remplie de 1 (sauf ligne i)

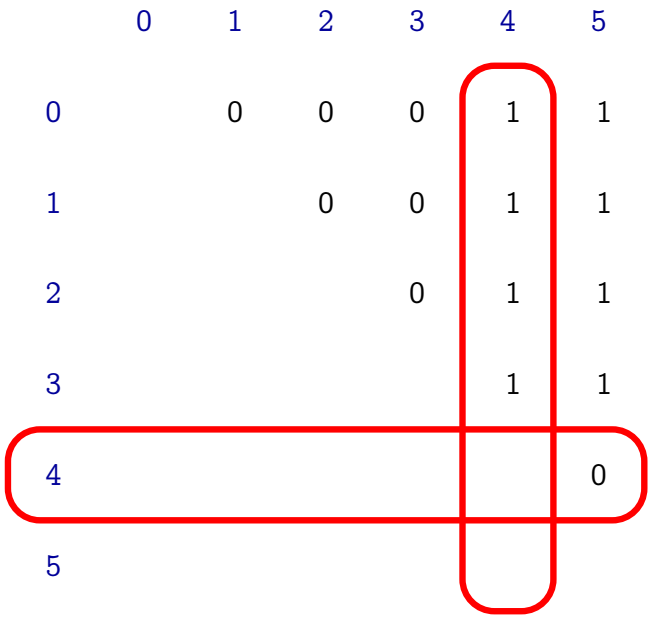
	0	1	2	3	4	5
0		0	0	0	1	1
1			0	0	1	1
2				0	1	1
3					1	1
4						0
5						

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

31	B1	2A	03	12	48

Implémentation

- Matrice triangulaire N = nombre de cadres
- Utilisation d'une page dans le cadre i
  $\rightarrow$ ligne i à 1 puis colonne i à 0
- Cadre le plus ancien (à utiliser) =
  ligne est remplie de 0, colonne remplie de 1 (sauf ligne i)



03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

31	B1	2A	03	76	48

Implémentation

- Matrice triangulaire N = nombre de cadres
- Utilisation d'une page dans le cadre i
  $\rightarrow$ ligne i à 1 puis colonne i à 0
- Cadre le plus ancien (à utiliser) =
  ligne est remplie de 0, colonne remplie de 1 (sauf ligne i)

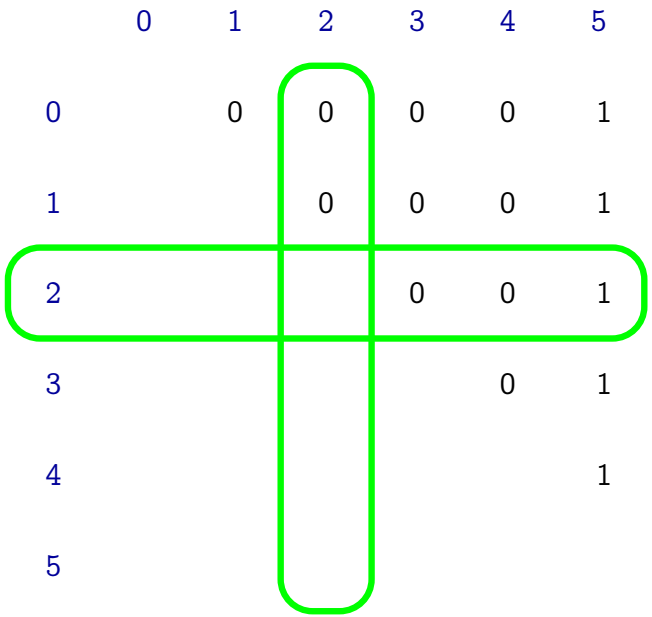
	0	1	2	3	4	5
0		0	0	0	0	1
1			0	0	0	1
2				0	0	1
3					0	1
4						1
5						

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

31	B1	2A	03	76	48

Implémentation

- Matrice triangulaire N = nombre de cadres
- Utilisation d'une page dans le cadre i
  $\rightarrow$ ligne i à 1 puis colonne i à 0
- Cadre le plus ancien (à utiliser) =
  ligne est remplie de 0, colonne remplie de 1 (sauf ligne i)



03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

31	B1	2A	03	76	48

Implémentation

- Matrice triangulaire N = nombre de cadres
- Utilisation d'une page dans le cadre i
  $\rightarrow$ ligne i à 1 puis colonne i à 0
- Cadre le plus ancien (à utiliser) =
  ligne est remplie de 0, colonne remplie de 1 (sauf ligne i)

	0	1	2	3	4	5
0		0	0	0	0	1
1			0	0	0	1
2				1	1	1
3					0	1
4						1
5						

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

31	B1	2A	03	76	48

Implémentation

- Matrice triangulaire N = nombre de cadres
- Utilisation d'une page dans le cadre i
  $\rightarrow$ ligne i à 1 puis colonne i à 0
- Cadre le plus ancien (à utiliser) =
  ligne est remplie de 0, colonne remplie de 1 (sauf ligne i)

	0	1	2	3	4	5
0		0	0	0	0	1
1			0	0	0	1
2				1	1	1
3					0	1
4						1
5						

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

31	B1	2A	03	76	1F

Implémentation

- Matrice triangulaire N = nombre de cadres
- Utilisation d'une page dans le cadre i
  $\rightarrow$ ligne i à 1 puis colonne i à 0
- Cadre le plus ancien (à utiliser) =
  ligne est remplie de 0, colonne remplie de 1 (sauf ligne i)

	0	1	2	3	4	5
0		0	0	0	0	0
1			0	0	0	0
2				1	1	0
3					0	0
4						0
5						

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

31	B1	2A	03	76	1F

LRU: implémentation

Implémentation

- Matrice triangulaire N = nombre de cadres
- Utilisation d'une page dans le cadre i
  $\rightarrow$ ligne i à 1 puis colonne i à 0
- Cadre le plus ancien (à utiliser) =

  ligne est remplie de 0, colonne remplie de 1 (sauf ligne i)

	0	1	2	3	4	5
0		0	0	0	0	0
1			0	0	0	0
2				1	1	0
3					0	0
4						0
5						

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

37	B1	2A	03	76	1F

Implémentation

- Matrice triangulaire N = nombre de cadres
- Utilisation d'une page dans le cadre i
  $\rightarrow$ ligne i à 1 puis colonne i à 0
- Cadre le plus ancien (à utiliser) =
  ligne est remplie de 0, colonne remplie de 1 (sauf ligne i)

	0	1	2	3	4	5
0		1	1	1	1	1
1			0	0	0	0
2				1	1	0
3					0	0
4						0
5						

03 04 2A 1F 2A 12 48 31 B1 2A 03 76 2A 1F 37

37	B1	2A	03	76	1F

Ce qu'il faut retenir

- Politique de remplacement de page
- FIFO (avec bit de seconde chance)
- LRU (implémentée au niveau matériel)
- Tracer l'exécution et compter les défauts