

Algorithme de remplacement de page

SYE

12 - Algorithme de remplacement de page

Abstract

Ce document traite des stratégies et algorithmes de remplacement de pages dans la gestion de la mémoire virtuelle. Il présente les approches globales et locales, détaillant les algorithmes OPT (optimal mais théorique), FIFO (simple mais peu efficace), et LRU (plus complexe mais performant en tenant compte de l'utilisation récente des pages). Des variantes comme l'algorithme de la seconde chance améliorent FIFO en utilisant des bits de référence pour prioriser les pages. Enfin, l'algorithme WSClock introduit le concept d'ensemble de travail (Working Set), optimisant la gestion des pages en mémoire grâce à une observation temporelle et un remplacement local basé sur les pages récemment utilisées dans un intervalle donné, avec des exemples concrets pour illustrer son fonctionnement.

Table des matières

| | |
|---|----------|
| 1. Notion de remplacement de page | 2 |
| 2. Algorithmes OPT, FIFO, LRU | 3 |
| 2.1. Algorithme OPT | 3 |
| 2.2. Algorithme FIFO | 3 |
| 2.3. Algorithme LRU | 4 |
| 3. Algorithme de la seconde chance | 5 |
| 4. Algorithme WSClock | 6 |
| 4.1. Notion d'ensemble de travail (Working Set) | 6 |
| 4.2. Algorithme | 6 |
| 4.2.1. Concept | 6 |
| 4.2.2. Étapes | 6 |
| 4.2.3. Avantages | 6 |
| 4.3. Exemple | 6 |

1. Notion de remplacement de page

Il existe deux stratégies de remplacement de page :

- **Remplacement global**
 - n'importe quelle page peut être remplacée
 - peut appartenir à n'importe quel processus
 - bonne stratégie globale
 - permet de préserver des pages du processus en cours d'exécution
- **Remplacement local**
 - seules les pages du processus en cours d'exécution peuvent être remplacées
 - permet de préserver des pages du processus en cours d'exécution
 - stratégie au niveau du processus seulement

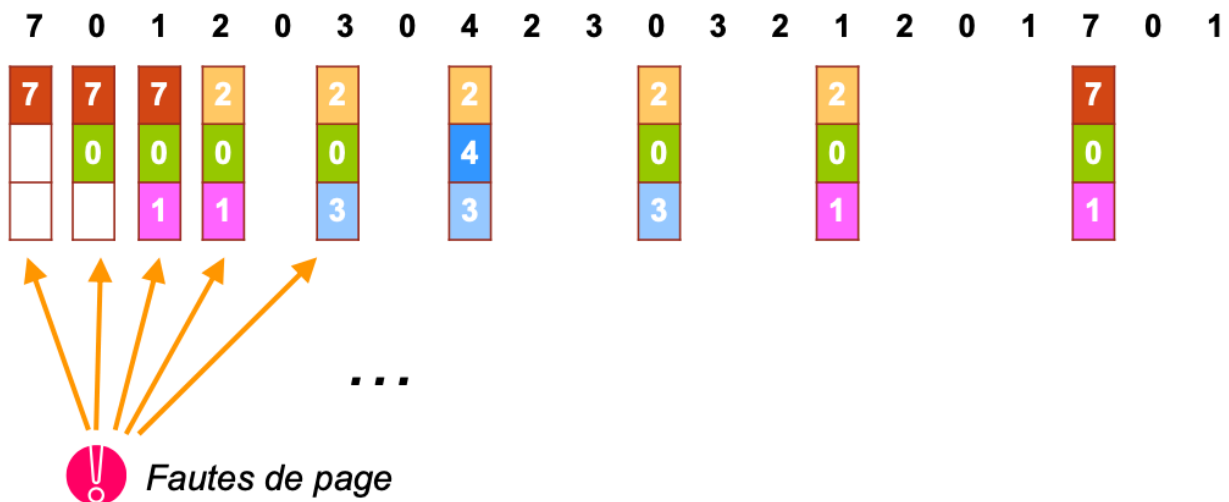
2. Algorithmes OPT, FIFO, LRU

2.1. Algorithme OPT

L'algorithme OPT est un algorithme de remplacement de page optimal. Il remplace la page qui ne sera pas utilisée pendant la plus longue période de temps. Cet algorithme est difficile à implémenter car il nécessite de connaître à l'avance les pages qui seront utilisées dans le futur.

Les premières pages provoquent forcément des fautes de pages, puisqu'aucune page n'est encore présente en mémoire.

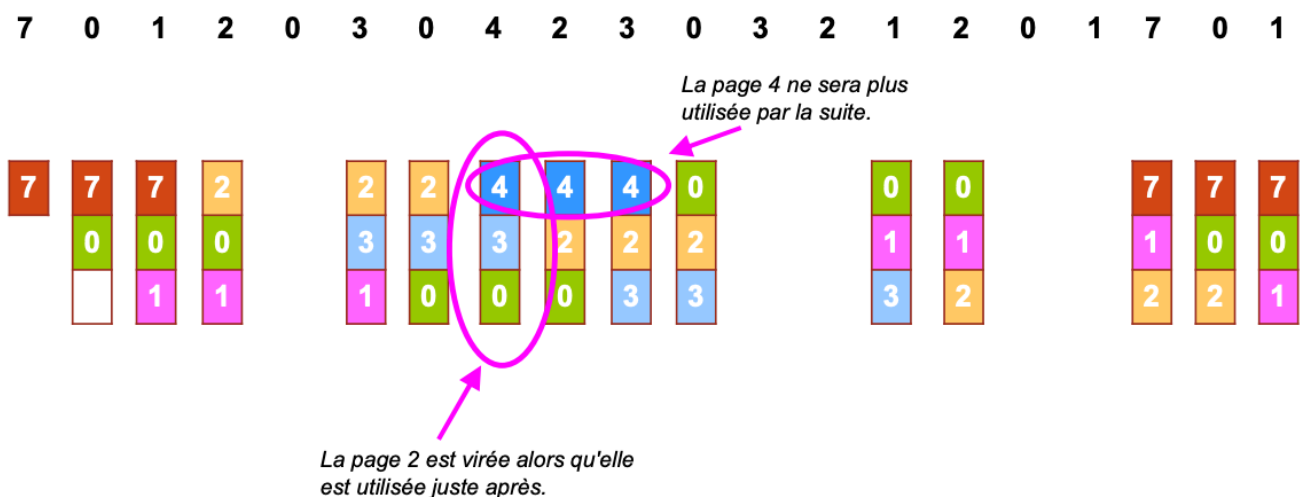
Séquence de références



2.2. Algorithme FIFO

L'algorithme FIFO est un algorithme de remplacement de page simple. Il remplace la page qui est en mémoire depuis le plus longtemps. Cet algorithme est facile à implémenter, mais il peut ne pas être optimal. Il ne tient pas en compte de l'utilisation récente et future des pages.

Séquence de références

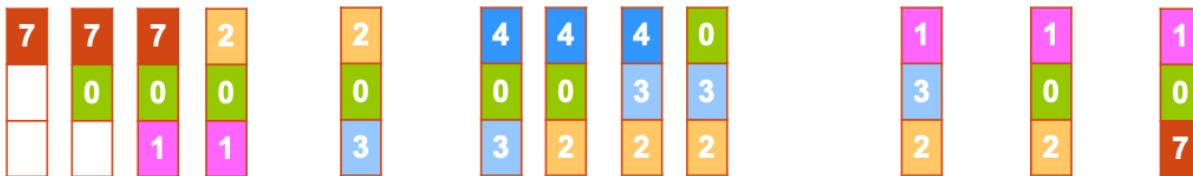


2.3. Algorithme LRU

L'algorithme LRU (Least Recently Used) est un algorithme de remplacement de page qui remplace la page qui n'a pas été utilisée depuis le plus longtemps. Cet algorithme est plus complexe à implémenter que l'algorithme FIFO, mais il est plus efficace. Il tient compte de l'utilisation récente des pages.

Séquence de références

7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1

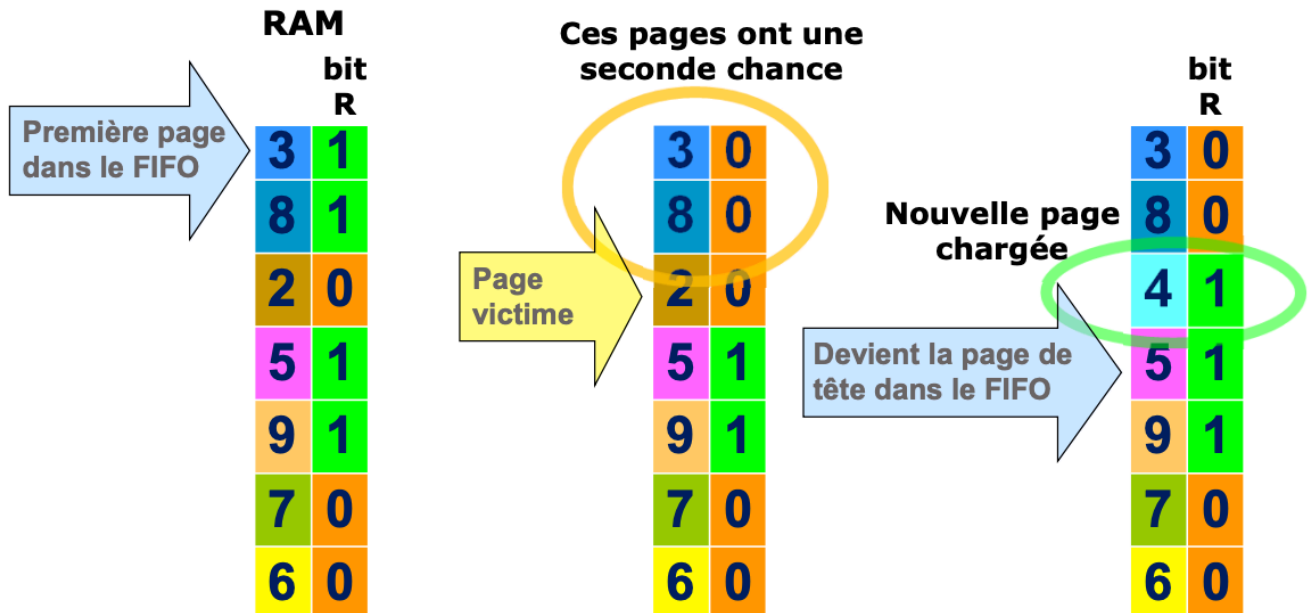


Pour l'implémenter le bit R de la page présenté dans le chapitre précédent est utilisé. Lorsqu'une page est utilisée, le bit R est mis à 1. Nous devons utiliser un timer pour remettre à 0 ce bit après un certain temps. On sélectionne la première page avec le bit R à 0.

3. Algorithme de la seconde chance

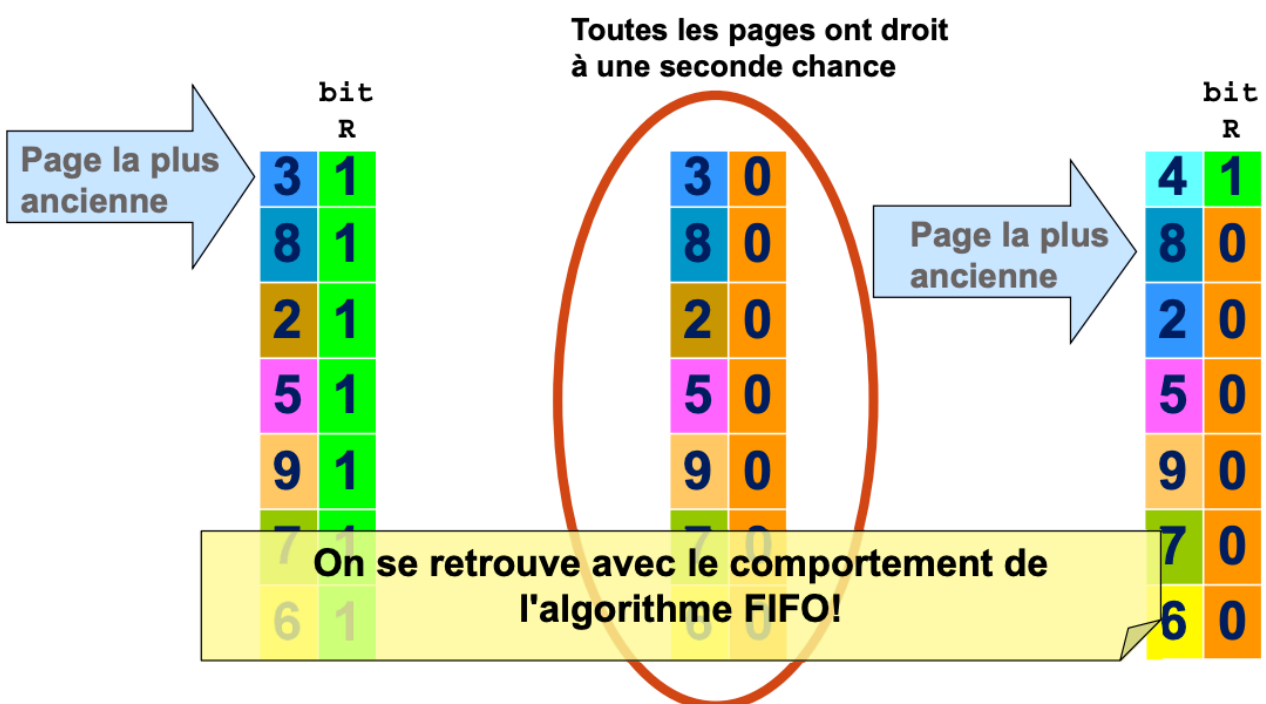
L'algorithme de la seconde chance est une amélioration de l'algorithme FIFO. Il utilise un bit R pour chaque page. Lorsqu'une page est utilisée, le bit R est mis à 1. Lorsqu'une page doit être remplacée, on vérifie le bit R. Si le bit R est à 1, la page est mise en fin de liste et le bit R est mis à 0. Sinon, la page est remplacée.

- Exemple: la page 4 est référencée...



Si toutes les pages ont le bit R à 1, on sélectionne la première page de la liste et on remet tous les bits R à 0. Cela revient à un fonctionnement de type FIFO.

- Autre exemple: la page 4 est référencée...



4. Algorithme WSClock

4.1. Notion d'ensemble de travail (Working Set)

- **Working Set** : Ensemble des pages d'un processus les plus récemment utilisées dans une fenêtre d'observation (Δ), exprimée en durée ou en nombre de pages.
- **Temps virtuel** : Temps perçu par le processus lorsqu'il est en exécution. Suspendu lorsque le processus n'est pas en cours d'exécution.
- Une page fait partie du Working Set si : $TVC - TDU \leq \Delta$, où TVC est le temps virtuel courant et TDU le temps de la dernière utilisation.
- TVC: Temps virtuel courant
- TDU: Temps de dernière utilisation

4.2. Algorithme

- Basé sur l'algorithme de la seconde chance et utilise un **remplacement local** (seules les pages du processus fautif peuvent être évincées).

4.2.1. Concept

- Associe un TDU (temps de dernière utilisation) à chaque page physique.
- TDU est mis à jour périodiquement pour les pages avec le bit de référence actif ($R = 1$).
- Les pages avec $R = 0$ ne sont pas mises à jour et sortent progressivement du Working Set.

4.2.2. Étapes

1. Parcourir les pages avec un pointeur.
2. Si $R = 1$, réinitialiser R à 0 et passer à la page suivante (seconde chance).
3. Si $R = 0$:
 - Si la page est dans le Working Set ($TVC - TDU \leq \Delta$), passer à la suivante.
 - Sinon, évincer la page.

4.2.3. Avantages

- Optimise l'utilisation des pages en mémoire.
- Permet un pré-paging en rechargeant tout le Working Set si le processus est réactivé.

4.3. Exemple

- **Configuration initiale** : $TVC = 32$, $\Delta = 10$.
- Pages examinées en boucle.
- Page "C" : $TVC - TDU = 32 - 24 = 8$, dans le Working Set (non remplacée).
- Page "D" : $TVC - TDU = 32 - 20 = 12$, hors du Working Set (remplacée).