

## 1 Algorithme de pagination

Soit la suite de références de pages virtuelles suivante :

1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5

On désire comparer les performances des algorithmes de gestion de remplacement de pages LRU (Least Recently Used) et FIFO (First In First Out).

**Question 1.1** *Combien y a-t-il de défauts de pages pour cette suite de référence avec 3 cadres mémoires respectivement pour :*

1. *l'algorithme FIFO;*
2. *l'algorithme LRU;*
3. *l'algorithme optimal ?*

**Question 1.2** *Même question avec 4 cadres.*

*N'y a-t-il pas des résultats surprenants ?*

**Question 1.3** *Répondre aux mêmes questions avec la suite de références suivante :*

3, 2, 4, 3, 4, 2, 2, 3, 4, 5, 6, 7, 7, 6, 5, 4, 5, 6, 7, 2, 1

## 2 Influence de l'écriture des programmes

En Java, comme en C, les tableaux bidimensionnels sont rangés en mémoire ligne par ligne. Soit la déclaration suivante :

```
int [][] tab = new int [1024][1024];
```

et les deux extraits de programmes :

```
for (int i = 0; i < 1024; i++)
    for (int j = 0; j < 1024; j++)
        tab[i] [j] = 1;
```

```
for (int j = 0; j < 1024; j++)
    for (int i = 0; i < 1024; i++)
        tab[i] [j] += 2;
```

**Question 2.1** *Combien de défauts de pages provoquent respectivement les deux doubles boucles précédentes, en supposant de pages de 4 Ko ?*

*Rappel : en Java les int sont stockés sur 4 octets.*

### 3 Working Set

Les pages victimes de l'algorithme de remplacement peuvent être prises globalement sur l'ensemble des processus ou seulement localement par processus ce qui permet de garantir un certain nombre de cadres disponibles par processus.

Le modèle de l'espace de travail (*Working Set*) dit qu'un processus doit avoir un nombre minimum de cadres à disposition sous peine de provoquer un nombre immodéré de défauts de pages<sup>1</sup>. Dès que la contention mémoire sera trop forte, on déchargera toutes les pages du processus. En contrepartie, on pourra recharger toutes les pages de l'espace de travail du processus lorsqu'il pourra reprendre son exécution effectuant ainsi un *préchargement*. Notons que la taille de l'espace de travail peut changer au cours de l'exécution d'un processus.

Le problème est d'évaluer l'espace de travail. Pour ce faire, le système se fixe une fenêtre d'observation temporelle et compte le nombre de références à des pages différentes pendant ce temps.

Soit la suite de références de pages virtuelles suivante :

3, 2, 4, 3, 4, 2, 2, 3, 4, 5, 6, 7, 7, 6, 5, 4, 5, 6, 7, 2, 1

**Question 3.1** *Calculez l'évolution de la taille de l'espace de travail pour des fenêtres d'observation de taille 6, 5 et 4.*

**Question 3.2** *Calculez le nombre de défauts de page dans chacun des cas avec FIFO comme algorithme de remplacement de pages.*

### 4 Question de tailles

**Question 4.1** *Pourquoi les tailles de page, le nombre de pages virtuelles et le nombre de cadres en mémoire centrale sont tous des puissances de 2 ?*

---

1. Ce qui peut conduire au problème de « trashing » où le processeur passe son temps à gérer la mémoire pour le compte du système plutôt que de travailler pour les processus utilisateurs.