جامعة هواري بومدين
للعلوم والتكنولوجيا
U S T H B

Série n°3

Gestion de la mémoire centrale

S. BOUKHEDOUMA

USTHB – FEI – département d'Informatique
Laboratoire des Systèmes Informatiques -LSI
sboukhedouma@usthb.dz

Série n°3 – Exercice 7

Exercice 7

On considère un système à mémoire paginée où la taille d'une page est de 200 mots. La mémoire centrale est constituée de 3 frames.
Soit la chaîne de références suivante :

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

Calculer le taux de défauts de pages générés en appliquant les algorithmes de remplacement **FIFO** et **Seconde Chance**.

Série n°3 – Exercice 7

Algorithmes de remplacement de pages

Algorithme FIFO
Nombre de frames = 3

La page la plus ancienne sera remplacée

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

Série n°3 – Exercice 7

	1	2	1	0	4	1	3	4	2	1	4	1
Frame 1	→1	→1	→1	→1	4	4	→4	→4	2	2	2	→2
Frame 2		2	2	2	→2	1	1	1	→1	→1	4	4
Frame 3				0	0	→0	3	3	3	3	→3	1
Défaut de page	D	D		D	D	D	D		D		D	D
Taux DP = 9/12 = 0,75 = 75%												

La flèche désigne le pointeur de tête de file

Série n°3 – Exercice 7

Algorithmes de remplacement de pages

Algorithme de la seconde chance

Utilise un bit de référencement R (initialisé à 1 et remis à 1 à chaque référencement)

La première page dont le bit R est à 0 est retirée (le parcours se fait de manière circulaire)

Une page visitée et non retirée (R=1) reçoit une deuxième chance (R devient à 0)

Si une page est référencée et se trouve déjà en mémoire, son bit R est mis à 1

Série n°3 – Exercice 7

Algorithme de la seconde chance

	1	2	1	0	4	1	3	4	2	1	4	1
Frame 1	→**1/1**	→1/1	→1/1	→1/1	**4/1**	4/1	→4/1	→4/1	**2/1**	**2/1**	→2/1	→2/1
Frame 2		**2/1**	2/1	2/1	→2/0	**1/1**	1/1	1/1	→1/0	→**1/1**	**1/0**	**1/1**
Frame 3				**0/1**	0/0	→0/0	**3/1**	3/1	3/0	3/0	**4/1**	4/1
Défaut de page	D	D		D	D	D	D		D		D	
Taux DP = 8/12 = 0,6666 = 66,66%												

<u>Pour charger la page 4</u>

L'algorithme parcourt les pages 1, 2 et 0 en mettant leur bit de référence R à 0
Retire la page 1 et la remplace par la page 4 (avec bit R =1)
Le pointeur de page avance sur la page 2

Série n°3 – Exercice 7

Algorithme de la seconde chance

	1	2	1	0	4	1	3	4	2	1	4	1
Frame 1	→**1/1**	→1/1	→1/1	→1/1	**4/1**	4/1	→4/1	→4/1	**2/1**	**2/1**	→2/1	→2/1
Frame 2		**2/1**	2/1	2/1	→2/0	**1/1**	1/1	1/1	→1/0	→**1/1**	**1/0**	**1/1**
Frame 3				**0/1**	0/0	→0/0	**3/1**	3/1	3/0	3/0	**4/1**	4/1
Défaut de page	D	D		D	D	D	D		D		D	
Taux DP = 8/12 = 0,6666 = 66,66%												

<u>Pour charger la page 1</u>

L'algorithme retire la page 2
Le pointeur de page avance sur la page 0

<u>Pour charger la page 2</u>

L'algorithme parcourt les pages 4, 1 et 3 en mettant leur bit de référence R à 0
Retire la page 4 et la remplace par la page 2 (avec bit R =1)
Le pointeur de page avance sur la page 1

Série n°3 – Exercice 7

Algorithme de la seconde chance

	1	2	1	0	4	1	3	4	2	1	4	1
Frame 1	→**1/1**	→1/1	→1/1	→1/1	**4/1**	4/1	→4/1	→4/1	**2/1**	**2/1**	→2/1	→2/1
Frame 2		**2/1**	2/1	2/1	→2/0	**1/1**	1/1	1/1	→1/0	→**1/1**	**1/0**	**1/1**
Frame 3				**0/1**	0/0	→0/0	**3/1**	3/1	3/0	3/0	**4/1**	4/1
Défaut de page	D	D		D	D	D	D		D		D	
Taux DP = 8/12 = 0,6666 = 66,66%												

<u>Au référencement de la page 1 (10ème et 12ème référence)</u>,
son bit de référence sera remis à 1

<u>Pour charger la page 4</u>
Le bit de référence de la page 1 est remis à 0
L'algorithme retire la page 3 et la remplace par la page 4
Le pointeur de page avance sur la page 2

Série n°3 – Exercice 7

Sur ce même système, soit le programme C suivant qui initialise une matrice carrée d'entiers A de taille 100x100:

```
for (int i = 0; i < 100; i++)
    for (int j = 0; j < 100; j++)
        A[i][j] = 0;
```

2. Sachant qu'un entier est sur un **mot** et que l'élément **A[0][0]** se trouve à l'adresse logique **200**, déduire le nombre de pages occupées par la matrice **A**. On suppose que ce programme ainsi que les variables **i** et **j** occupent la **page 0** du processus.

3. Déterminer la chaine de références correspondant à l'exécution de ce programme.

Série n°3 – Exercice 7

2. Nombre de pages de la matrice

Soit

1 mot = entier

Taille de la matrice = 100*100 mots = 10^4 mots

Taille d'une page = 200 mots

Nombre de pages occupées par la matrice = $10^4/200$

Nb=50 pages

Série n°3 – Exercice 7

2. Nombre de pages de la matrice

Nombre de pages occupées par la matrice = $10^4/200$

Nb=50 pages

Une ligne i de la matrice occupe 100 mots

Organisation des lignes en mémoire virtuelle

Les lignes 0 et 1 sont rangées à la page 1
Les lignes 2 et 3 sont rangées à la page 2
…
Les lignes 98 et 99 sont rangées à la page 50

Série n°3 – Exercice 7

3. La chaine de références

```
for (int i = 0; i < 100; i++)
        for (int j = 0; j < 100; j++)
            A[i][j] = 0;
```

Le programme est chargé à la page 0 de la mémoire
Les données (la matrice) est chargé dans les pages 1 2 3 4 ... 50

Toute instruction du programme fait référence à la page 0
Tout accès aux variables i et j fait référence à la page 0
Tout accès à un élément de la matrice fait référence à l'une des pages 1 2 3 4 ... 50

Série n°3 – Exercice 7

3. La chaine de références

```
for (int i = 0; i < 100; i++)
        for (int j = 0; j < 100; j++)
            A[i][j] = 0;
```

Programme équivalent

```
i = 0
while (i<100)
    { j = 0;
      while (j<00)
          { A[i][j] = 0; j++; }
i++; }
```

Série n°3 – Exercice 7

3. La chaine de références

Les pages référencées

```
i = 0                    → page 0
while (i<100)            → page 0
    { j = 0;             → page 0
      while (j<00)       → page 0
          { A[i][j] = 0; → page 0 + une autre page (1, 2, …, 50) en fonction de
                           l'élément de la matrice
            j++; }       → page 0

      i++; }             → page 0
```

Série n°3 – Exercice 7

3. La chaine de références

L'instruction

A[i][j] = 0; → page 0 + une autre page (1, 2, …, 50) en fonction de l'élément de la matrice

1ere ligne : 100 fois la séquence 0 1 (0 1 0 1 0 1 …)

A[0][0] = 0; → page 0 + page 1

A[0][1] = 0; → page 0 + page 1

…

A[0][99] = 0; → page 0 + page 1

2eme ligne: 100 fois la séquence 01 (0 1 0 1 0 1…)

A[1][0] = 0; → page 0 + page 1

A[1][99] = 0; → page 0 + page 1

…

→ 200 fois la séquence de pages 0 1

Série n°3 – Exercice 7

3. La chaine de références

Partie initialisation : 4 fois le référencement à la page 0 : 0 0 0 0 → 0

Pour tout élément A[0][j] = 0 : référencement des pages 0 1
A chaque incrémentation de j et condition (j <100) : 2 fois le référencement à la page 0 : 0 0 → 0
Donc pour la première ligne on a la chaine : 0 1 0 (100 fois)

Incrémentation de i et condition (i <100) : 2 fois le référencement à la page 0 : 0 0 → 0
Idem pour la deuxième ligne on a la chaine : 0 1 0 (100 fois)

Pour les autres lignes …

3. La chaine de références

Pour tout élément A[2][j] = 0 : référencement des pages 0 2
A chaque incrémentation de j et condition (j <100) : 2 fois le référencement à la page 0 : 0 0 → 0
Donc pour la première ligne on a la chaine : 0 2 0 (100 fois)

Incrémentation de i et condition (i <100) : 2 fois le référencement à la page 0 : 0 0 → 0
Idem pour la deuxième ligne on a la chaine : 0 2 0 (100 fois)

Ainsi de suite
Pour les lignes 3 et 4 : référencement des pages 0 3 0 (200 fois)
Pour les lignes 5 et 6 : référencement des pages 0 4 0 (200 fois)
…
Pour les lignes 98 et 99 : référencement des pages 0 50 0 (200 fois)

Série n°3 – Exercice 7

3. La chaine de références

Pour récapituler,

0 0 1 0 0 2 0 0 3 0 0 4 0 … 0 50 0

200 fois 200 fois 200 fois 200 fois 200 fois

on devra tirer une chaine de référence réduite (en prenant en compte que le nombre de frames libres est au moins égal à deux)

→ **la chaine réduite est:**

0 1 0 2 0 3 0 4 0 5 0 … 49 0 50 0

Série n°3 – Exercice 7

4. Considérant que la **page 0** est déjà chargée dans le **frame 0** de la mémoire centrale et que les deux autres frames sont initialement libres, quel est le nombre de défauts de pages engendrés par l'application de l'algorithme de remplacement **LRU** sur les **14** premières références de la chaine de références ?

5. Déduire le nombre total de défauts de pages engendrés par l'application de l'algorithme de remplacement **LRU** sur toute la chaine de références.

Série n°3 – Exercice 7

Les 14 premières références de la chaine de référence

0 1 0 2 0 3 0 4 0 5 0 6 0 7

Algorithmes de remplacement de pages

Algorithme LRU

La page la plus loin dans le passé (dans la chaine de références) sera remplacée

Série n°3 – Exercice 7

Algorithme LRU

On suppose que la page 0 est déjà chargée dans le frame 0

	0	1	0	2	0	3	0	4	0	5	0	6	0	7
Frame 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Frame 1		1	1	1	1	3	3	3	3	5	5	5	5	7
Frame 2				2	2	2	2	4	4	4	4	6	6	6
Défaut de page		D		D		D		D		D		D		D
Taux DP = 7/14 = 0,5 = 50%														

Série n°3 – Exercice 7

Algorithme LRU sur la chaine globale

On a 7 défauts de pages sur les 14 premières références

La chaine de référence entière comporte : 101 références alternées entre 0 et une autre page (1 2 3 …. 50)

Sur les deux frames libres une page remplace une autre
La page 0 est résidente en mémoire (n'est pas remplacée par l'algorithme)

Par déduction le nombre de défauts de pages sur la chaine complète est égal à 50

→ **taux de DP est égal à 50%**

Série n°3 – Exercice 7

Question supplémentaire

On considère les temps d'accès suivants:

Ta1: temps d'accès à une page déjà chargée en MC

Ta2: temps d'accès à une page non chargée en MC avec existence d'un frame libre ou page remplacée non modifiée

Ta3: temps d'accès à une page (aucun frame libre et page remplacée modifiée)

Quel est le temps d'accès effectif (Tae) pour l'exécution de la chaine réduite ? (la page 0 est déjà en mémoire initialement)

Tae = 51*Ta1+ 2*Ta2 + 48*Ta3