

9 - REMPLACEMENT DE PAGES

1. ALGORITHME OPTIMAL

Lors d'un défaut de page, on remplace la page qui sera utilisée le plus tard possible. Il est clair que cet algorithme n'est pas réalisable car il faudrait pour cela connaître l'avenir; cependant, il est d'une grande utilité pour étalonner les autres algorithmes.

On suppose qu'un processus fait référence à ses pages dans l'ordre suivant:

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

1.1.

Si on ne limite pas le nombre de cases allouées au processus, combien de défauts de page sont provoqués par cette suite d'accès ? Combien de cases au minimum faut-il allouer au processus pour atteindre ce nombre ?

Au minimum, il y a autant de défauts de page que de chargements, soit 6. Pour atteindre ce nombre, il ne faut jamais décharger une page que l'on sera amené à recharger par la suite. Lorsqu'on charge la page 0, on peut remplacer la page 5 qui ne sera jamais réutilisée.

Par contre, la page 1 ne peut pas être chargée à la place de la page 0 qui sera réutilisée après le chargement de 1, 2 ne peut être chargée ni à la place de 0, ni à la place de 1, etc. Il faut donc 5 cases pour une exécution avec un minimum de défauts de page.

1.2.

Donnez l'évolution de la table des pages ainsi que le nombre de défauts de pages si on alloue au processus 3 cases.

Il y a 8 défauts de pages.

	5	0	1	2	0	3	0	4	2	3	0	3	2	1	2
0		2						X			2				
1			3			X								3	
2				1											
3						3								X	
4								2			X				
5	1			X											
	D	D	D	D		D		D			D			D	

2. ALGORITHME FIFO

Lors d'un défaut de page, on remplace la plus ancienne page qui ait été chargée.

2.1.

On suppose qu'un processus fait référence à ses pages dans l'ordre suivant:

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

Donnez l'évolution de la table des pages ainsi que le nombre de défauts de pages si on alloue au processus 3 cases.

Cet exercice est fait pour montrer le nombre de défauts de page de l'algorithme FIFO. sur la suite de références aux pages qui est utilisée dans la suite du TD.

3 cases	5	0	1	2	0	3	0	4	2	3	0	3	2	1	2
0		2				X	3			X	1				
1			3				X							2	
2				1				X	2					X	3
3						2			X	3					X
4								1			X				
5	1			X											
	D	D	D	D		D	D	D	D	D	D			D	D

Il y a 12 défauts de pages.

3. ALGORITHME FINUFO (FIRST IN NOT USED FIRST OUT)

On suppose qu'il y a un bit R dans le descriptif de chaque page. A chaque accès à une page, le matériel met le bit R à 1. Lorsqu'une page est chargée, le bit R est aussi mis à 1.

Une liste des pages triées selon leur date de chargement est gérée : la page en tête de liste est la plus récemment chargée, celle en fin de liste est la plus anciennement chargée. Lors d'un défaut de page, la page la plus anciennement chargée est examinée. Si son bit R vaut 1, elle est remise en tête de liste et son bit R passe à 0 (elle dispose ainsi d'une deuxième chance). La page déchargée est donc celle qui est la plus anciennement chargée et dont le bit R vaut 0.

3.1.

On suppose qu'un processus fait référence à ses pages dans l'ordre suivant:

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

Donnez l'évolution de la table des pages ainsi que le nombre de défauts de pages si on alloue au processus 3 cases.

Il y a 11 défauts de pages.

	5	0	1	2	0	3	0	4	2	3	0	3	2	1	2
	5,1	0,1	1,1	2,1	2,1	3,1	3,1	4,1	2,1	2,1	0,1	3,1	3,1	1,1	2,1
		5,1	0,1	1,0	1,0	0,0	0,1	3,0	4,1	4,1	2,0	0,1	0,1	3,0	1,1
			5,1	0,0	0,1	2,1	2,1	0,0	3,0	3,1	4,0	2,0	2,1	0,0	3,0
0		2			2		2		X		3				X
1			3			X								2	
2				1				X	2				2	X	3
3						3				3	X	1			
4								1				X			
5	1			X											
	D	D	D	D		D		D	D		D	D		D	D

4. ALGORITHME LRU (LEAST RECENTLY USED)

Lors d'un défaut de page, on remplace la page la moins récemment utilisée.

4.1.

On suppose qu'un processus fait référence à ses pages dans l'ordre suivant:

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

Dans le cas où on alloue au processus 3 cases, donnez l'évolution de la liste des pages chargées classées par ordre de dernière utilisation. Quel est le nombre de défauts de pages ?

Il y a 10 défauts de pages.

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

5. ALGORITHME NRU (NOT RECENTLY USED)

On suppose qu'il y a deux bits R et M dans le descriptif de chaque page. A chaque accès à une page, le matériel met le bit R à 1. De plus, si la page est modifiée, il met aussi le bit M à 1. Lorsqu'une page est chargée, le bit R est mis à 1 et la valeur du bit M est déterminée par le type d'accès qui a provoqué le chargement. Périodiquement, tous les bits R sont remis à zéro (par exemple sur les tops d'une horloge toutes les 20 ms). Les pages peuvent ainsi être classées suivant quatre catégories:

Catégorie 0: R = 0, M = 0.

Catégorie 1: R = 0, M = 1.

Catégorie 2: R = 1, M = 0.

Catégorie 3: R = 1, M = 1.

Lors d'un défaut de page, parmi les pages de plus petite catégorie, on remplace la plus anciennement chargée.

5.1.

On suppose qu'un processus fait référence à ses pages dans l'ordre suivant:

5E 0L 1L 2E 0E 3E 0L 4E 2L 3L 0L 3L 2E 1E 2L

La lettre indiquant si l'accès se fait en lecture ou en écriture. En supposant qu'un top d'horloge intervient tous les cinq accès mémoire, donnez l'évolution de la table des pages ainsi que le nombre de défauts de pages si on alloue au processus 3 cases.

Le tableau suivant contient le numéro de case ainsi que les valeurs des bits R et M.

Il y a 11 défauts de pages.

	5E	0L	1L	2E	0E	top	3E	0L	4E	2L	3L	top	0L	3L	2E	1E	2L
0		2,10		X	3,11	3,01		3,11		X			3,10			X	
1			3,10		X											3,11	
2				2,11		2,01			X	3,10		3,00	X		2,11		2,11
3							1,11				1,11	1,01		1,11			
4									2,11			2,01			X		
5	1,11					1,01	X										
	D	D	D	D	D		D		D	D			D		D	D	

6. ALGORITHME PAR GESTION DE FENETRE (WS : WORKING SET)

Dans cet algorithme une fenêtre (i.e. l'espace de travail ou *working set*) est un intervalle de la suite des accès aux pages. Le nombre de pages contenues dans la fenêtre varie en fonction des accès. Périodiquement, la fenêtre est mise à jour en ne conservant que les pages utilisées durant la période.

6.1.

On suppose qu'un processus fait référence à ses pages dans l'ordre suivant:

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

En supposant que la fenêtre est mise à jour tous les 3 accès mémoire (on ne conserve donc dans la fenêtre que les pages accédées lors des 3 dernières références), donnez l'évolution de la liste des pages chargées. Quel est le nombre de défauts de pages ?

1	D	5													
2	D	5	0												
3	D	5	0	1											
4	D	5	0	1	2										
5		5		1	2	0									
6	D	5		1	2	0	3								
7					2		3	0							
8	D				2		3	0	4						
9						3	0	4	2						
10	D						0	4	2	3					
11								4	2	3	0				
12								4	2		0	3			
13	D										0	3	2		
14	D										0	3	2	1	
15											0	3		1	2

Il y a 9 défauts de pages.