

Module : Système d'exploitation 1

Correction de la Série n°3 : Gestion de la mémoire centrale

Exercice 1 :

Schéma mémoire après allocation :

60ko
32ko
75ko
15ko
18ko libre

Mémoire usager de 200ko

Après allocation aux 4 premiers programmes, il reste 18 ko de la mémoire usager, ce qui est insuffisant pour 30ko

Cas First-Fit

30ko
17ko
13ko libre
32ko
75ko libre
15ko
18ko libre

Cas Best-Fit

30ko
30ko libre
32ko
32ko
75ko libre
15ko
17ko
1ko libre

Cas worst-Fit

17ko
43ko libre
32ko
30ko
45ko libre
15ko
18ko libre

Exercice 2

Taille page=1ko, la mémoire usager=4ko=4 cadres de pages sont disponibles pour l'exécution des programmes.

Taille programme=8ko

1. Couple (p,d) associés aux références :

P=adr/taille page, d=adr modulo taille page. Exp=p=1/1024=0 d=1 modulo 1204=1

1, 2076, 85, 1500, 3648, 100, 4314, 1025, 89, 5741, 1219, 4500, 7658, 4096, 6999, 7191, 5140, 128 →

(0,1),(2,28), (0,85), (1,476), (3,572), (0,100), (4,218), (1,1),(0,89),(5,621),(1,195),(4,404),
(7,490),(4,0),(6,855),(7,23),(5,20),(0,128).

La chaîne de références est donc : CR=0,2,0,1,3,0,4,1,0,5,1,4,7,4,6,7,5,0.

2. Le taux de défaut de page : Mc=4 pages

- **Fifo** : on choisit la page la plus ancienne. Chaque fois que l'on charge une page en MC, on enfile le numéro de cette page dans une file gérée en FIFO.

0,2,0,1,3,0,4,1,0,5,1,4,7,4,6,7,5,0.																			
CR	0	2	0	1	3	0	4	1	0	5	1	4	7	4	6	7	5	0	Total DP
M C	P0	0	0	0	0	0	0	4	4	4	4	4	7	7	7	7	7	0	
	P1	libre	2	2	2	2	2	2	0	0	0	0	4	4	4	4	4	4	
	P2	libre	libre	libre	1	1	1	1	1	1	5	5	5	5	5	6	6	6	
	P3	libre	libre	libre		3	3	3	3	3	3	1	1	1	1	1	1	5	5
Défaut de page (DP)	1	1	0	1	1	0	1	0	1	1	1	0	1	1	1	0	1	1	13

Le taux de défaut de page = Nombre DP/Taille CR = 13/18 = 0.72 = 72%

- **LRU** : choisit la page qui n'a pas été utilisée depuis longtemps

0,2,0,1,3,0,4,1,0,5,1,4,7,4,6,7,5,0.																			
CR	0	2	0	1	3	0	4	1	0	5	1	4	7	4	6	7	5	0	Total DP
M C	P0	0	0	0	0	0	0	0	0	0	0	0	7	7	7	7	7	7	
	P1	libre	2	2	2	2	2	4	4	4	4	4	4	4	4	4	4	0	
	P2	libre	libre	libre	1	1	1	1	1	1	1	1	1	1	1	1	5	5	
	P3	libre	libre	libre	libre	3	3	3	3	3	5	5	5	5	5	6	6	6	
Défaut de page (DP)	1	1	0	1	1	0	1	0	0	1	0	0	1	0	1	0	1	1	10

Le taux de défaut de page = Nombre DP/Taille CR = 10/18 = 0.55 = 55%

- **OPT**: la page qui ne sera pas référencée durant le plus grand temps à venir

0,2,0,1,3,0,4,1,0,5,1,4,7,4,6,7,5,0.																			
CR	0	2	0	1	3	0	4	1	0	5	1	4	7	4	6	7	5	0	Total DP
M C	P0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	P1	libre	2	2	2	2	4	4	4	4	4	4	4	4	6	6	6	6	
	P2	libre	libre	libre	1	1	1	1	1	1	1	1	7	7	7	7	7	7	
	P3	libre	libre	libre		3	3	3	3	3	5	5	5	5	5	5	5	5	
Défaut de page (DP)	1	1	0	1	1	0	1	0	0	1	0	0	1	0	1	0	0	0	08

Le taux de défaut de page = Nombre DP/Taille CR = 08/18 = 0.44 = 44%

L'algorithme OPT minimise le taux de défaut de page.

Exercice 3

Cette technique s'appelle : gestion de la mémoire virtuelle par segmentation (voir le cours)

(1,45) est une adresse virtuelle de la forme (s,d) tel que : s : numéro du segment virtuel et d : déplacement dans le segment s virtuel.

Donc (1,45) : s=1 et d=45 → voir la table des segments sur la ligne numéro de segment=1 et adresse mémoire=2048 → donc pour un déplacement d=45 ,on l'ajoute à l'adresse mémoire 2048 → $2048+45=2093$

les adresses physiques correspondantes aux adresses virtuelles suivantes : (1,45),(0,200),(1,468),(2,115),(3,56),(4,600),(4,1012) sont respectivement :

2093-878-2516-179-3304-1598-2010

Exercice4 :

a. Le mode de partage de mémoire utilisé est **la segmentation paginée**, puisque les programmes sont composés de segments, et l'espace d'adressage mémoire divisé en pages.

b. Le programme est composé de :

Segment de code de 32768 octets, qui nécessite 8 pages virtuelles de code.

Segment données 5 pages dont la dernière ne contiendra que 2 octets.

Segment pile 4 pages dont la dernière ne contiendra que 3582 octets.

Le programme nécessite 17 pages, qui **ne rentrera pas** dans l'espace d'adressage alloué à un processus, soit de 16 pages.

c. La même méthode que b si la page est de taille 8 octets