

**TD N° 03 : Gestion de la mémoire****Questions**

- a) Rappeler brièvement le principe du swapping, de la pagination et de la segmentation
- b) Citez les 2 types de fragmentation communément rencontrés et le type de systèmes auxquels ils se rapportent.
- c) Fragmentation interne → système paginés, fragmentation externe → systèmes segmentés.

**Exercice 01**

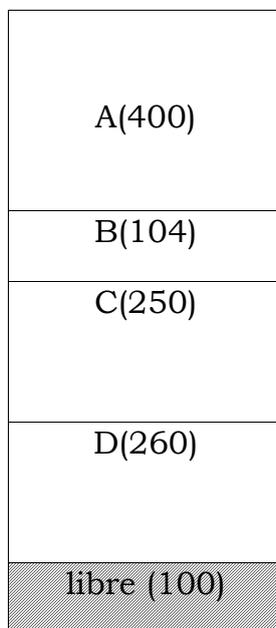
Etant donné un système de gestion mémoire en partitions variables, La taille de mémoire utilisateur est de 1114 KO. L'utilisateur veut exécuter les 5 programmes suivants (A, 400 KO), (B, 104 KO), (C, 250 KO), (D, 260), (E, 120 KO)

- a- Donner le schéma de mémoire obtenu après l'allocation, que déduisez vous?
- b- Les programmes A, B, C sont terminés et l'utilisateur relance le programme E et un programme (F, 100 KO), Quels est le schéma mémoire après allocation selon:
- b-1) First Fit (première partition libre)
  - b-2) Worst Fit (plus grande partition libre)
  - b-3) Best Fit (meilleure partition libre)

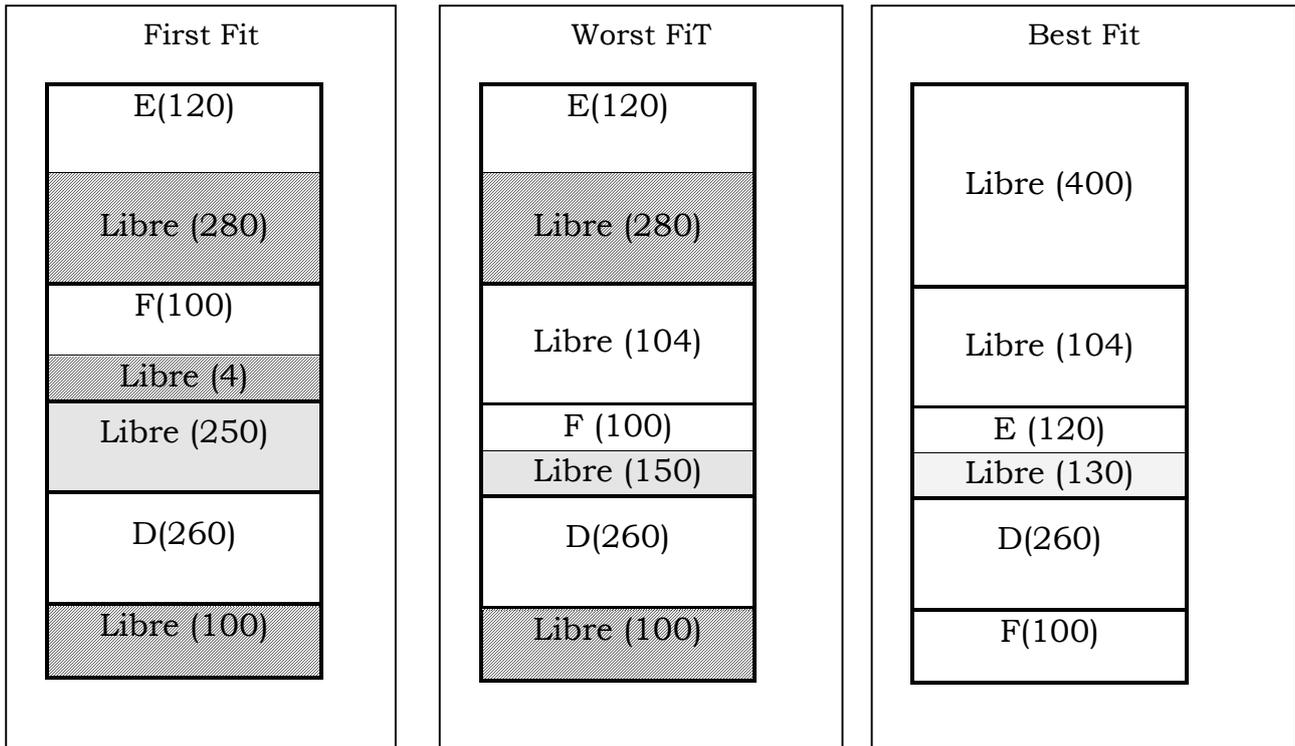
**Solution Exo 01**

Gestion des partitions variables de la mémoire de taille 1114 KO,

- a- schéma de mémoire après allocation



Le programme E(120) ne peut pas être exécuté (mémoire insuffisante)



**Exercice 02**

Soi un système de pagination dont la taille de page est de 1KO, la mémoire usager est de 4KO

1- Combien de cadres (pages physiques) sont initialement disponibles pour l'exécution des processus?

2- Soit un programme de taille 8KO, qui fait référence aux adresses logiques: 1, 2076, 85, 1500, 3648, 100, 4314, 1025, 89, 5741, 1219, 4500, 7658, 4096, 6999, 7191, 5140, 128

Donner le couple (p, d) associé à chaque référence mémoire, en déduire la chaine de référence associée?

3- Donner le taux de défaut de page induit par un remplacement FIFO, LRU, seconde chance, quel est le meilleur algorithme?

**Solution Exo 02**

1- 4 cadres sont disponibles pour l'exécution des programmes.

2- Taille du programme est de 8KO

(p, d)=? p=adresse *div* taille page      d= adresse *modulo* taille page

Donc, pour chaque adresse:

(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 chaine de référence {0, 2, 0, 1, 3, 0, 4, 1, 0, 5, 1, 4, 7, 4, 6, 7, 5, 0 }

3- Algorithme FIFO

Chaîne		0	2	0	1	3	0	4	1	0	5	1	4	7	4	6	7	5	0
Cadres en MC	P <sub>0</sub>	0	0	0	0	0	0	4	4	4	4	4	4	7	7	7	7	7	0
	P <sub>1</sub>		2	2	2	2	2	2	2	0	0	0	0	0	4	4	4	4	4
	P <sub>2</sub>				1	1	1	1	1	1	5	5	5	5	5	6	6	6	6
	P <sub>3</sub>					3	3	3	3	3	3	1	1	1	1	1	1	5	5
Défaut Page		X	X		X	X		X		X	X	X		X	X	X		X	X

Nombre de défauts de page: 13

Le taux de DP= 13/18 = 72.22%

*Algorithme LRU*

Chaîne		0	2	0	1	3	0	4	1	0	5	1	4	7	4	6	7	5	0	
Cadres en MC	P <sub>0</sub>	0	0	0	0	0	0	0	0	0	0	0	0	7	7	7	7	7	7	
	P <sub>1</sub>		2	2	2	2	2	4	4	4	4	4	4	4	4	4	4	4	0	
	P <sub>2</sub>				1	1	1	1	1	1	1	1	1	1	1	1	1	1	5	5
	P <sub>3</sub>					3	3	3	3	3	3	5	5	5	5	5	6	6	6	6
Défaut Page		X	X		X	X		X			X			X		X		X	X	

Nombre de défauts de page: 10

Le taux de DP= 10/18 = 55.55%

*Algorithme seconde chance*

Chaîne		0	2	0	1	3	0	4	1	0	5	1	4	7	4	6	7	5	0
Cadres en MC	P <sub>0</sub>	0 <sub>0</sub>	0 <sub>0</sub>	0 <sub>1</sub>	0 <sub>1</sub>	0 <sub>1</sub>	0 <sub>1</sub>	0 <sub>0</sub>	0 <sub>0</sub>	0 <sub>1</sub>	0 <sub>0</sub>	0 <sub>0</sub>	0 <sub>0</sub>	7 <sub>0</sub>	7 <sub>0</sub>	7 <sub>0</sub>	7 <sub>1</sub>	7 <sub>1</sub>	7 <sub>1</sub>
	P <sub>1</sub>		2 <sub>0</sub>	4 <sub>1</sub>	4 <sub>1</sub>	4 <sub>1</sub>	4 <sub>0</sub>	4 <sub>0</sub>	4 <sub>0</sub>	0 <sub>0</sub>									
	P <sub>2</sub>				1 <sub>0</sub>	1 <sub>0</sub>	1 <sub>0</sub>	1 <sub>0</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>0</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>1</sub>	1 <sub>0</sub>	1 <sub>0</sub>	5 <sub>0</sub>	5 <sub>0</sub>
	P <sub>3</sub>					3 <sub>0</sub>	5 <sub>0</sub>	6 <sub>0</sub>	6 <sub>0</sub>	6 <sub>0</sub>	6 <sub>0</sub>								
Défaut Page		X	X		X	X		X			X			X		X		X	X

Nombre de défauts de page: 10

Le taux de DP= 10/18 = 83.33%

meilleur algorithme est LRU.

**Exercice 03**

On considère la table des segments suivante pour un processus P1 :

Segment	Base	Limite
0	540	234
1	1254	128
2	54	328
3	2048	1024
4	976	200

1) Calculez les adresses réelles correspondant aux adresses virtuelles suivantes (vous signalerez éventuellement les erreurs d'adressage) :

(0:128), (1:100), (2:465), (3:888), (4:100), (4:344)

2) L'adresse virtuelle (4,200) est-elle valide ?

- (0:128) : déplacement valide ( $128 < 234$ ).  $\text{Adr\_physique} = \text{base} + \text{limite} = 540 + 128 = \mathbf{668}$ .
- (1:100) : déplacement valide ( $100 < 128$ ).  $\text{Adr\_physique} = \text{base} + \text{limite} = 1254 + 100 = \mathbf{1354}$ .
- (2:465) : déplacement invalide ( $465 > 328$ ).
- (3:888) : déplacement valide ( $888 < 1024$ ).  $\text{Adr\_physique} = \text{base} + \text{limite} = 2048 + 888 = \mathbf{2936}$ .
- (4:100) : déplacement valide ( $100 < 200$ ).  $\text{Adr\_physique} = \text{base} + \text{limite} = 976 + 100 = \mathbf{1076}$ .
- (4:344) : déplacement invalide car ( $344 > 200$ )

2) Non. Dans un segment de longueur 200, les déplacements valides sont dans l'intervalle [0-199].