

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 1

Exercice 1

A). En appliquant une stratégie de placement **First-Fit**, représentez l'évolution de la mémoire

centrale, suite à chacun des événements suivants :

- 1) arrivée du programme **G** de taille égale à 20 Ko;
- 2) départ du programme **B**;
- 3) arrivée du programme **H** de taille égale à 15 Ko;
- 4) départ du programme **E**;
- 5) arrivée du programme **I** de taille égale à 40 Ko.

B). Même question, mais cette fois-ci, on applique une stratégie de placement **Best-Fit** et **Worst-Fit**.

L'état initial d'occupation de la mémoire est donné dans la figure suivante:

Série n°3– Exercice 1

Etat initial de la mémoire

A (10K)	10K	B (20K)	30K	C (10K)	5K	D (30K)	10K	E (5K)	15K	F (20K)	10 K
------------	-----	------------	-----	------------	----	---------	-----	-----------	-----	---------	------

A, B, C, D, E, F sont déjà chargés en mémoire

Les tailles des partitions libres sont indiquées en couleur orange

Série n°3– Exercice 1

A (10K)	10K	B (20K)	30K	C (10K)	5K	D (30K)	10K	E (5K)	15K	F (20K)	10 K
------------	-----	------------	-----	------------	----	---------	-----	-----------	-----	---------	------

Stratégie « First-Fit » : choisir la première partition qui convient pour charger un programme

Série n°3– Exercice 1

A (10K)	10K	B (20K)	30K	C (10K)	5K	D (30K)	10K	E (5K)	15K	F (20K)	10 K
------------	-----	------------	-----	------------	----	---------	-----	-----------	-----	---------	------

arrivée de G (20K) → chargement de G

A (10K)	10K	B (20K)	G (20K)	10K	C (10K)	5K	D (30K)	10K	E (5K)	15K	F (20K)	10 K
------------	-----	------------	---------	-----	------------	----	---------	-----	-----------	-----	---------	------

Fin de B → libération de 20K → création d'une partition libre de 30K

A (10K)	30K	G (20K)	10K	C (10K)	5K	D (30K)	10K	E (5K)	15K	F (20K)	10 K
------------	-----	---------	-----	------------	----	---------	-----	-----------	-----	---------	------

Série n°3– Exercice 1

arrivée de H (15K) → chargement de H

A (10K)	H (15K)	15K	G (20K)	10K	C (10K)	5K	D (30K)	10K	E (5K)	15K	F (20K)	10 K
------------	------------	-----	---------	-----	------------	----	---------	-----	-----------	-----	---------	------

Fin de E → libération de 5K → création d'une partition libre de 30K

A (10K)	H (15K)	15K	G (20K)	10K	C (10K)	5K	D (30K)	30K		F (20K)	10 K
------------	------------	-----	---------	-----	------------	----	---------	-----	--	---------	------

arrivée de I (40 K) → Pas de partition de taille suffisante pour charger I

Série n°3– Exercice 1

Problème de fragmentation externe

Solution : compactage de la mémoire

A (10K)	H (15K)	G (20K)	C (10K)	D (30K)	F (20K)	70K
------------	---------	---------	------------	---------	---------	-----

Chargement de I (40K) → partition libre de 30K

A (10K)	H (15K)	G (20K)	C (10K)	D (30K)	F (20K)	I (40K)	30 K
------------	---------	---------	------------	---------	---------	---------	---------

Série n°3 – Exercice 1

A (10K)	10K	B (20K)	30K	C (10K)	5K	D (30K)	10K	E (5K)	15K	F (20K)	10 K
------------	-----	------------	-----	------------	----	---------	-----	-----------	-----	---------	------

Stratégie « Best-Fit » : choisir la plus petite partition qui convient pour charger un programme → laisser le plus petit résidu possible dans une partition

Série n°3– Exercice 1

A (10K)	10K	B (20K)	30K	C (10K)	5K	D (30K)	10K	E (5K)	15K	F (20K)	10 K
------------	-----	------------	-----	------------	----	---------	-----	-----------	-----	---------	------

arrivée de G (20K) → chargement de G

A (10K)	10K	B (20K)	G (20K)	10K	C (10K)	5K	D (30K)	10K	E (5K)	15K	F (20K)	10 K
------------	-----	------------	---------	-----	------------	----	---------	-----	-----------	-----	---------	------

Fin de B → libération de 20K → création d'une partition libre de 30K

A (10K)	30K	G (20K)	10K	C (10K)	5K	D (30K)	10K	E (5K)	15K	F (20K)	10 K
------------	-----	---------	-----	------------	----	---------	-----	-----------	-----	---------	------

Série n°3– Exercice 1

arrivée de H (15K) → chargement de H dans **la plus petite** partition qui convient

A (10K)	30K	G (20K)	10K	C (10K)	5K	D (30K)	10K	E (5K)	H (15 K)	F (20K)	10 K
------------	-----	---------	-----	------------	----	---------	-----	-----------	-------------	---------	------

Fin de E → libération de 5K → création d'une partition libre de 15K

A (10K)	30K	G (20K)	10K	C (10K)	5K	D (30K)	15K	H (15 K)	F (20K)	10 K
------------	-----	---------	-----	------------	----	---------	-----	-------------	---------	------

arrivée de I (40 K) → Pas de partition de taille suffisante pour charger I

Série n°3– Exercice 1

Comme la stratégie « First Fit » : Problème de fragmentation externe

Solution : compactage de la mémoire

A (10K)	G(20K)	C (10K)	D (30K)	H (15 K)	F (20K)	70K
------------	--------	------------	---------	-------------	---------	-----

Chargement de I (40K) → partition libre de 30K

A (10K)	G (20K)	C (10K)	D (30K)	H (15 K)	F (20K)	I (40K)	30K
------------	---------	------------	---------	-------------	---------	---------	-----

Série n°3 – Exercice 1

A (10K)	10K	B (20K)	30K	C (10K)	5K	D (30K)	10K	E (5K)	15K	F (20K)	10 K
------------	-----	------------	-----	------------	----	---------	-----	-----------	-----	---------	------

Stratégie « Worst-Fit » : choisir la plus grande partition qui convient pour charger un programme → laisser le plus grand résidu possible dans une partition

Série n°3– Exercice 1

A (10K)	10K	B (20K)	30K	C (10K)	5K	D (30K)	10K	E (5K)	15K	F (20K)	10 K
------------	-----	------------	-----	------------	----	---------	-----	-----------	-----	---------	------

arrivée de G (20K) → chargement de G

A (10K)	10K	B (20K)	G (20K)	10K	C (10K)	5K	D (30K)	10K	E (5K)	15K	F (20K)	10 K
------------	-----	------------	---------	-----	------------	----	---------	-----	-----------	-----	---------	------

Fin de B → libération de 20K → création d'une partition libre de 30K

A (10K)	30K	G (20K)	10K	C (10K)	5K	D (30K)	10K	E (5K)	15K	F (20K)	10 K
------------	-----	---------	-----	------------	----	---------	-----	-----------	-----	---------	------

Série n°3– Exercice 1

arrivée de H (15K) → chargement de H

A (10K)	H (15K)	15K	G (20K)	10K	C (10K)	5K	D (30K)	10K	E (5K)	15K	F (20K)	10 K
------------	------------	-----	---------	-----	------------	----	---------	-----	-----------	-----	---------	------

Fin de E → libération de 5K → création d'une partition libre de 30K

A (10K)	H (15K)	15K	G (20K)	10K	C (10K)	5K	D (30K)	30K		F (20K)	10 K
------------	------------	-----	---------	-----	------------	----	---------	-----	--	---------	------

arrivée de I (40 K) → Pas de partition de taille suffisante pour charger I

Série n°3– Exercice 1

Problème de fragmentation externe

Solution : compactage de la mémoire

A (10K)	H (15K)	G (20K)	C (10K)	D (30K)	F (20K)	70K
------------	---------	---------	------------	---------	---------	-----

Chargement de I (40K) → partition libre de 30K

A (10K)	H (15K)	G (20K)	C (10K)	D (30K)	F (20K)	I (40K)	30 K
------------	---------	---------	------------	---------	---------	---------	---------

Série n°3– Exercice 1

Commentaires

- Dans ce cas précis, la stratégie « **Worst-Fit** » se comporte exactement comme la stratégie « **First-Fit** »
- Aucune des trois stratégies n'est meilleure que l'autre car elles conduisent à un **problème de fragmentation externe**
- Nécessité d'une **opération de compactage** mémoire qui s'avère très coûteuse car il s'agit de reloger les programmes et un re-calcul d'adresses

Série n°3 – Exercice 2

Exercice 2

Considérons un système d'allocation mémoire à partitions variables.

Soient des partitions mémoire libres de 50k, 30k, 200k, 16k, et 30k (dans cet ordre).

1. Comment chacun des algorithmes « First-fit », « Best-fit », et « Worst-fit » placerait-il des processus de tailles respectives : 20k, 30k, 10k, 100k, et 60k ?
2. Quel algorithme effectue l'utilisation la plus efficace de la mémoire ?

Série n°3– Exercice 2

Stratégie First-Fit : choisir la première partition qui convient pour charger un programme

Etat initial

50K		30K		200K		16K		30K
-----	--	-----	--	------	--	-----	--	-----

Série n°3– Exercice 2

Chargement de P1

P1 (20K)	30K		30K		200K		16K		30K
-------------	-----	--	-----	--	------	--	-----	--	-----

Chargement de P2

P1 (20K)	P2 (30K)		30K		200K		16K		30K
-------------	-------------	--	-----	--	------	--	-----	--	-----

Chargement de P3

P1 (20K)	P2 (30K)		P3 10K	20K		200K		16K		30K
-------------	-------------	--	-----------	-----	--	------	--	-----	--	-----

Série n°3– Exercice 2

Chargement de P4

P1 (20K)	P2 (30K)		P3 10K	20K		P4 (100K)	100K		16K		30K
-------------	-------------	--	-----------	-----	--	-----------	------	--	-----	--	-----

Chargement de P5

P1 (20K)	P2 (30K)		P3 10K	20K		P4 (100K)	P5 (60K)	40K		16K		30K
-------------	-------------	--	-----------	-----	--	-----------	-------------	-----	--	-----	--	-----

Série n°3– Exercice 2

Stratégie « Best-Fit »: choisir la plus petite partition qui convient pour charger un programme → laisser le plus petit résidu possible dans une partition

Etat initial

50K		30K		200K		16K		30K
-----	--	-----	--	------	--	-----	--	-----

Série n°3– Exercice 2

Etat initial

50K		30K		200K		16K		30K
-----	--	-----	--	------	--	-----	--	-----

Chargement de P1

50K		P1 20K	10K		200K		16K		30K
-----	--	-----------	-----	--	------	--	-----	--	-----

Chargement de P2

50K		P1 20K	10K		200K		16K		P2 (30K)
-----	--	-----------	-----	--	------	--	-----	--	----------

Chargement de P3

50K		P1 20K	P3 (10K)		200K		16K		P2 (30K)
-----	--	-----------	-------------	--	------	--	-----	--	----------

Série n°3– Exercice 2

Chargement de P4

50K		P1 20K	P3 (10K)		P4 (100K)	100K		16K		P2 (30K)
-----	--	-----------	-------------	--	-----------	------	--	-----	--	-------------

Chargement de P5

50K		P1 20K	P3 (10K)		P4 (100K)	P5 (60K)	40K		16K		P2 (30K)
-----	--	-----------	-------------	--	-----------	----------	-----	--	-----	--	-------------

Série n°3– Exercice 2

Stratégie « Worst-Fit »: choisir la plus grande partition qui convient pour charger un programme → laisser le plus grand résidu possible dans une partition

Etat initial



Série n°3– Exercice 2

Etat initial

50K		30K		200K		16K		30K
-----	--	-----	--	------	--	-----	--	-----

Chargement de P1

50K		30K		P1 (20K)	180K		16K		30K
-----	--	-----	--	-------------	------	--	-----	--	-----

Chargement de P2

50K		30K		P1 (20K)	P2 (30K)	150K		16K		30K
-----	--	-----	--	-------------	-------------	------	--	-----	--	-----

Chargement de P3

50K		30K		P1 (20K)	P2 (30K)	P3 10K	140K		16K		30K
-----	--	-----	--	-------------	-------------	-----------	------	--	-----	--	-----

Série n°3– Exercice 2

Chargement de P4

50K		30K		P1 (20K)	P2 (30K)	P3 10K	P4 (100K)	40K		16K		30K
-----	--	-----	--	-------------	-------------	-----------	-----------	-----	--	-----	--	-----

Chargement de P5 impossible : Pb de **fragmentation externe**

Série n°3– Exercice 2

Commentaires

- Dans ce cas précis, la stratégie « **Worst-Fit** » n'est pas bonne car elle génère un **problème de fragmentation externe**.
- La stratégie « **Best Fit** » génère **moins de résidus** dans les partitions mémoires, donc elle semble être meilleure que la stratégie « **First Fit** ».

Série n°3– Exercice 2

Etat final avec « Best Fit »

50K		P1 20K	P3 (10K)		P4 (100K)	P5 (60K)	40K		16K		P2 (30K)
-----	--	-----------	-------------	--	-----------	----------	-----	--	-----	--	-------------

Etat final avec « First Fit »

P1 (20K)	P2 (30K)		P3 10K	20K		P4 (100K)	P5 (60K)	40K		16K		30K
-------------	-------------	--	-----------	-----	--	-----------	-------------	-----	--	-----	--	-----

Série n°3 – Exercice 3

Exercice 3

Considérons la table de segments suivante :

Segment	Base	Limite
0	1219	600
1	3300	14
2	90	100
3	2327	580
4	1952	96

Calculer les adresses physiques associées aux adresses logiques suivantes :

a. 0, 430

b. 1, 15

c. 2, 50

d. 3, 400

e. 4, 112

Série n°3 – Exercice 3

Dans une mémoire segmentée, l'adresse virtuelle (logique) est donnée par le couple (s, d)

s: numéro du segment

d: déplacement dans le segment

L'@ physique = @début du segment + déplacement dans le segment

@début du segment (base) est donnée dans la table des segments

La valeur de d (déplacement) doit être inférieure ou égale à la limite du segment

Série n°2 – Exercice 3

- a. 0, 430 b. 1, 15 c. 2, 50 d. 3, 400 e. 4, 112

@logique (0, 430)

@ physique = base (segment 0) + déplacement
= 1219 + 430 = **1649**

@logique (1, 15)

@ physique = base (segment 1) + déplacement

@ logique erronée car

limite (segment 1) = 14 < déplacement =15

Série n°3 – Exercice 3

a. 0, 430 b. 1, 15 c. 2, 50 d. 3, 400 e. 4, 112

@logique (2, 50)

@ physique = base (segment 2) + déplacement
= 90 + 50 = **140**

@logique (3, 400)

@ physique = base (segment 3) + déplacement
= 2327 + 400 = **2727**

@logique (4, 112)

@ physique = base (segment 4) + déplacement

@ logique erronée car

limite (segment 4) = 96 < déplacement = 112

Série n°3 – Exercice 4

Exercice 4

Dans un système paginé, les pages font 256 mots mémoire et on autorise chaque processus à utiliser au plus 4 cadres de la mémoire centrale. On considère la table des pages suivante du processus P1 (ici, les numéros de cadres (frames) sont donnés en binaire):

Page	0	1	2	3	4	5	6	7
Cadre	011	001	000	010	100	111	101	110
Présence	oui	non	oui	non	non	non	oui	non

- 1) Quelle est la taille de l'espace d'adressage du processus P1 ?
- 2) De combien de mémoire vive dispose ce système ?

Série n°3 – Exercice 4

Exercice 4

Page	0	1	2	3	4	5	6	7
Cadre	011	001	000	010	100	111	101	110
Présence	oui	non	oui	non	non	non	oui	non

3) Calculez les adresses réelles correspondant aux adresses virtuelles suivantes (vous signalerez

Eventuellement, les erreurs d'adressage) :

240, 546, 1578, 2072

4) Que se passe-t-il si P1 génère l'adresse virtuelle 770 ?

5) On considère l'adresse virtuelle suivante: 0000 0000 0000 0111. Sachant que les 4 bits de poids fort désigne le numéro de page et que 12 bits suivants représentent le déplacement dans la page, donnez l'adresse physique exprimée en binaire) correspondant à cette adresse.