1

Chapitre 5: GESTION DU MÉMOIRE CENTRALE

Dr M.CHENAIT (mchenait@usthb.dz)

Introduction.

Dr M.CHENAIT
(mchenait@usthb.dz)

- ✓ La relation entre ce chapitre et ces prédécesseurs.
- ✓ Le but du chapitre...

Dr M.CHENAIT
(mchenait@usthb.dz)

Le but du chapitre est de connaitre

Le gestionnaire de la mémoire doit répondre aux questions suivantes :

- Comment organiser la mémoire ? (une ou plusieurs partitions, le nombre et la taille des partitions fixes ou variables au cours du temps).

- Faut-il allouer une zone contiguë à chaque processus ou non ?

- Comment mémoriser l'état de la mémoire?

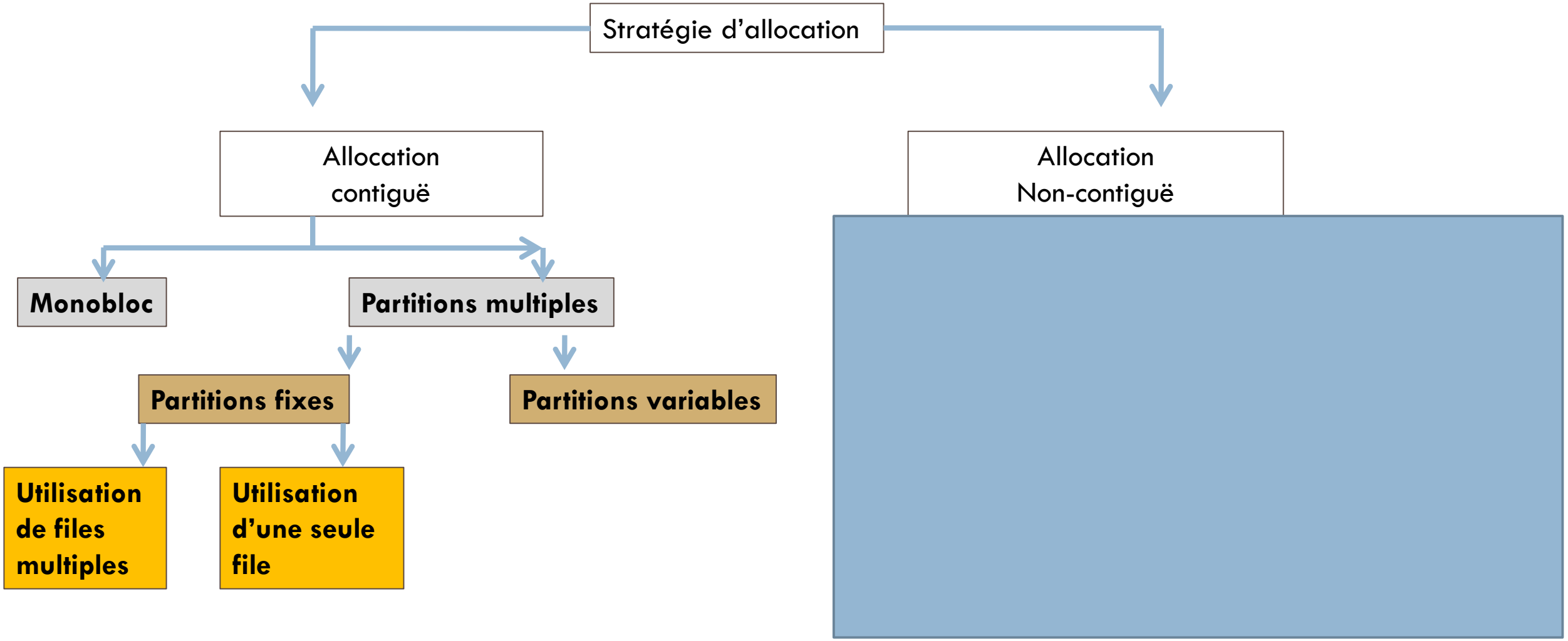
- S'il n'y a pas assez d'espace en mémoire, doit-on libérer de l'espace en retirant des parties ou des processus entiers? Si oui lesquels ?

- Les adresses figurant dans les instructions sont-elles relatives? Si oui, comment les convertir en adresses physiques.

- Si plusieurs processus peuvent être résidents en mémoire, comment assurer la protection des processus (éviter qu'un processus soit altéré par un autre?)

Stratégie d'allocation (politiques)

Dr M.CHENAIT
(mchenait@usthb.dz)

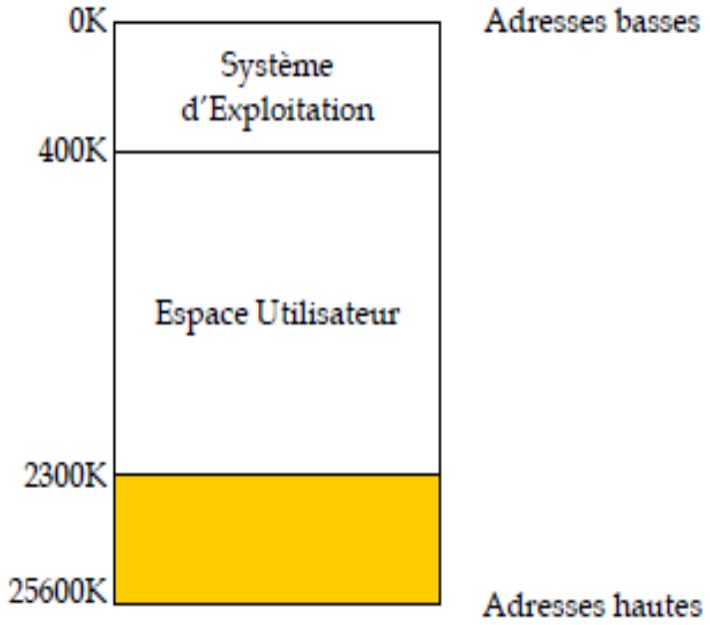


Stratégie d'allocation

Dr M.CHENAIT
(mchenait@usthb.dz)

Monobloc :

Dans ce cas, la mémoire est subdivisée en **deux partitions contiguës, une pour le système**
d'exploitation résidant en mémoire basse avec le vecteur d'interruptions et
l'autre pour le processus utilisateur. Elle n'autorise qu'un seul processus actif en mémoire à un
instant donné dont tout l'espace mémoire usager lui est alloué.

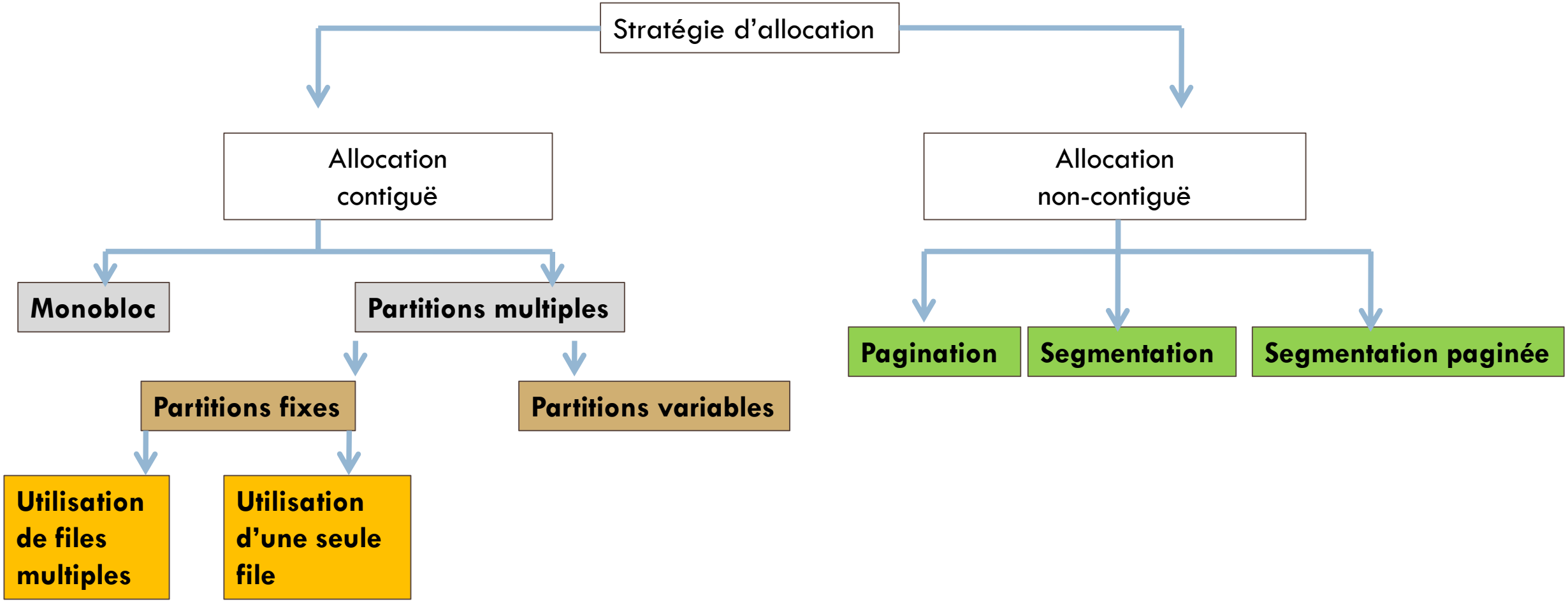


Avantage: Première solution

Inconvénient: La sous utilisation (i.e. mauvaise utilisation) de
la mémoire.

Stratégie d'allocation

Dr M.CHENAIT
(mchenait@usthb.dz)

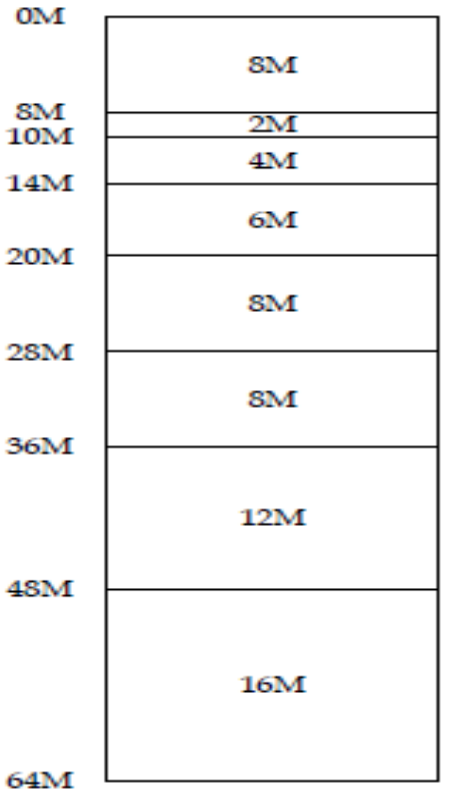


Stratégie d'allocation

Dr M.CHENAIT
(mchenait@usthb.dz)

Partitions multiples fixes:

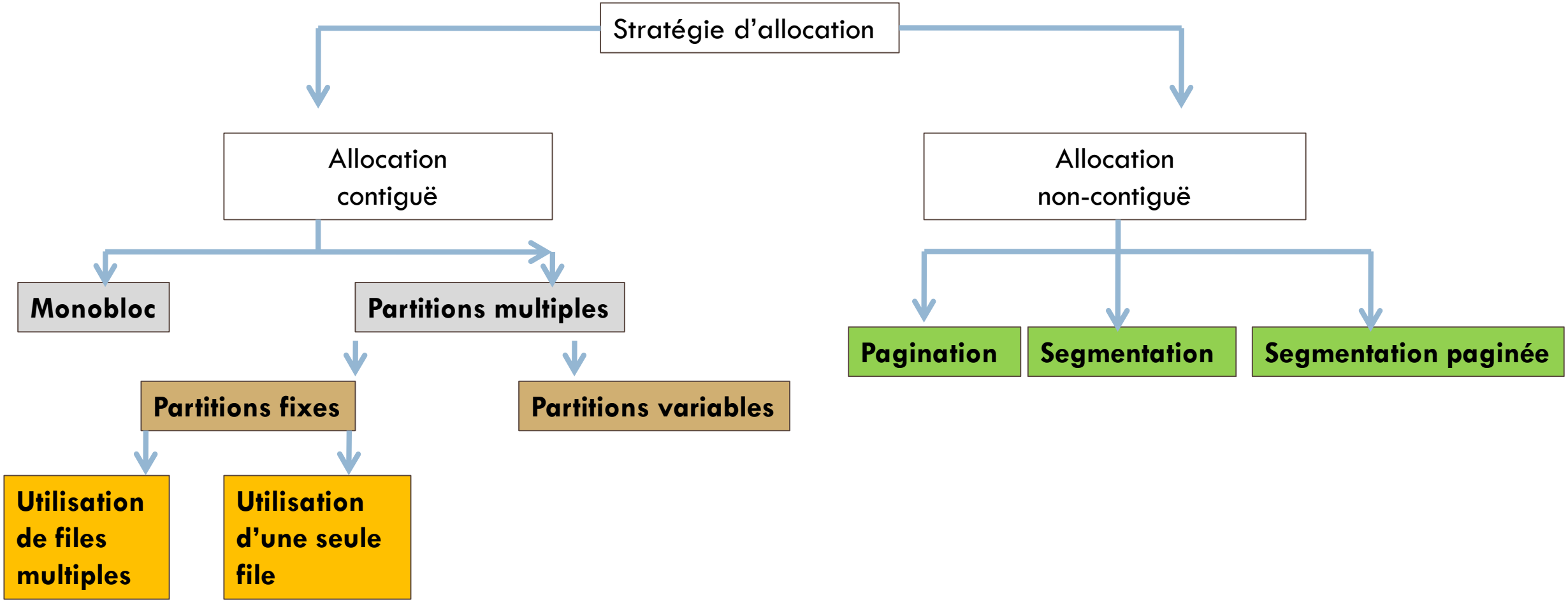
Cette solution consiste à diviser la mémoire en de partitions fixes, de tailles **pas nécessairement** égales, à l'initialisation du système. Le système d'exploitation maintient une table de description des partitions (PDT) indiquant les parties de mémoire disponibles (holes) et celles qui sont occupées.
Les programmes n'ayant pu se loger en mémoire sont placés dans une **file d'attente** (une file unique ou une file par partition)



Partitions de tailles inégales

Stratégie d'allocation

Dr M.CHENAIT
(mchenait@usthb.dz)



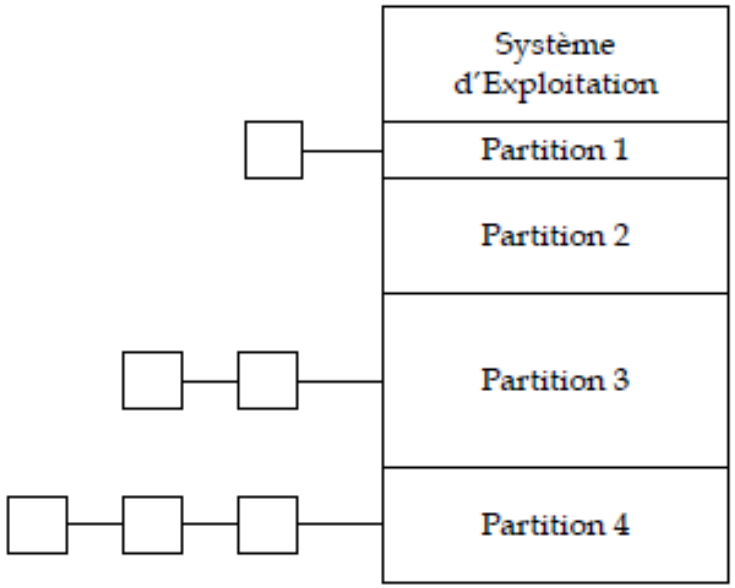
Stratégie d'allocation

Dr M.CHENAIT
(mchenait@usthb.dz)

Partitions multiples fixes (à files multiples)

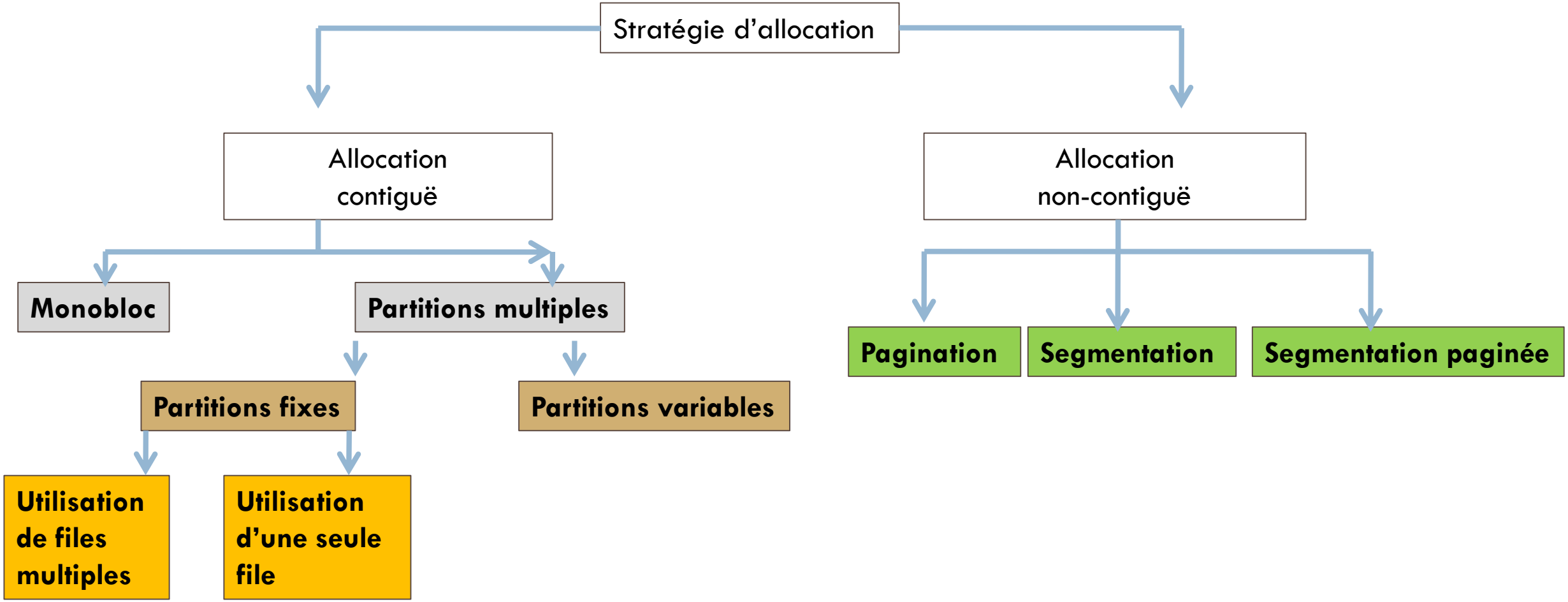
Chaque nouveau processus est placé dans la file d'attente de la plus petite partition qui peut le contenir.

Avantages: Plusieurs allocation à la fois.
Inconvénients: Les inconvénients de cette stratégie est qu'on perd en général de la place au sein de chaque partition et aussi, il peut y avoir des partitions inutilisées (leur file d'attente est vide).



Stratégie d'allocation

Dr M.CHENAIT
(mchenait@usthb.dz)

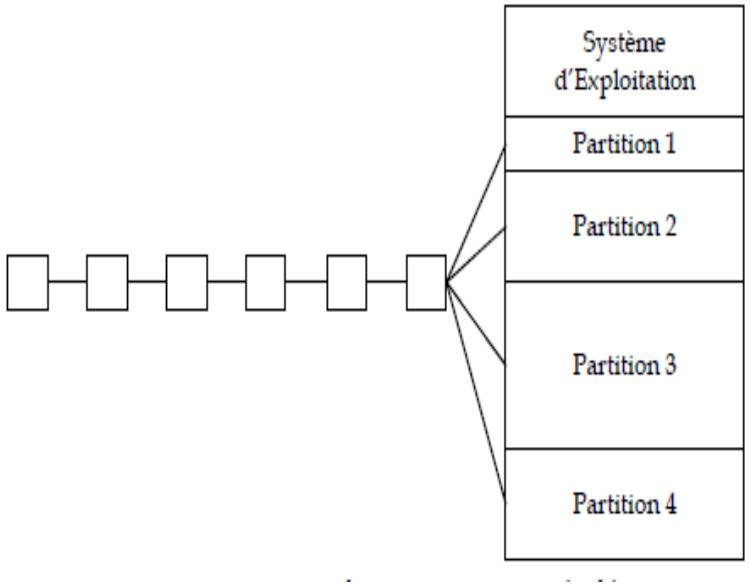


Stratégie d'allocation

Dr M.CHENAIT
(mchenait@usthb.dz)

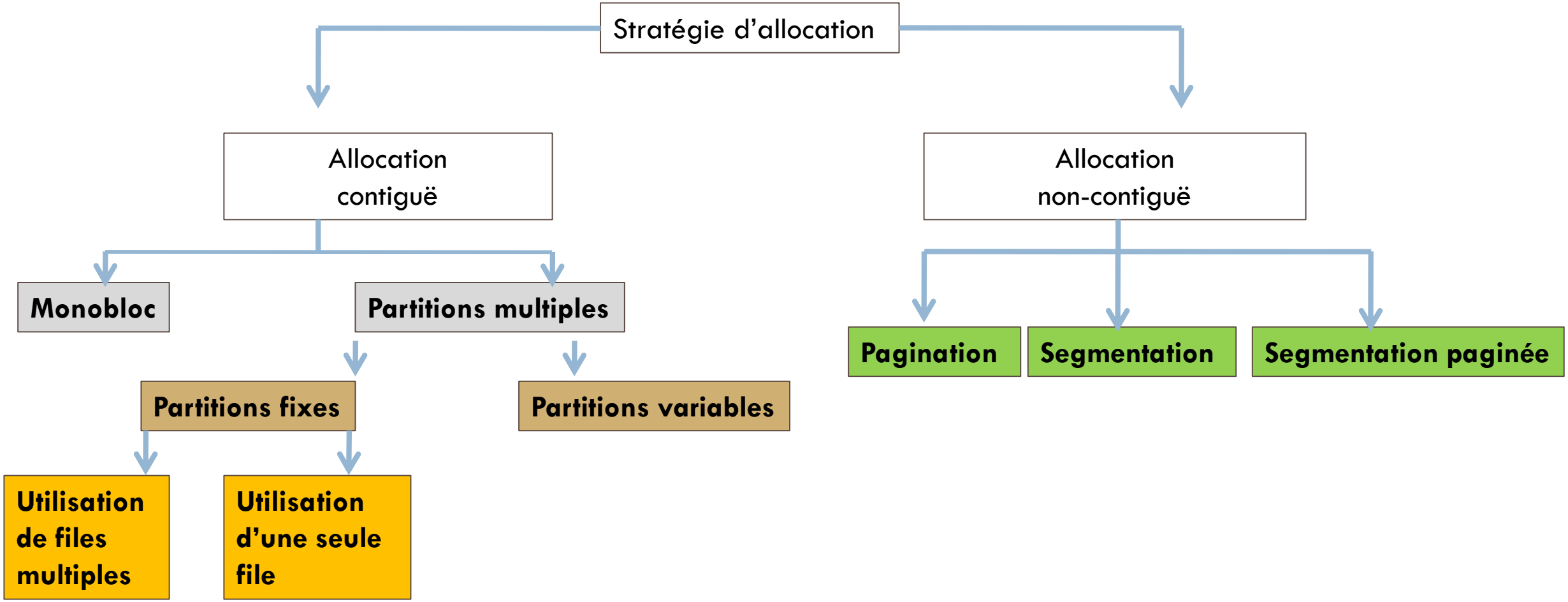
- **Partitions multiples fixes (à une files)**

Deux stratégies d'attribution d'une partition libre ;
1. Dès qu'une partition se libère, on lui affecte le **premier processus de la file qui** peut y tenir. **L'inconvénient** est qu'on peut ainsi affecter une partition de grande taille à un petit processus.
2. Dès qu'une partition se libère, on lui affecte le **plus grand processus de la file** qui peut y tenir. L'inconvénient est qu'on pénalise les processus de petite taille.



Stratégie d'allocation

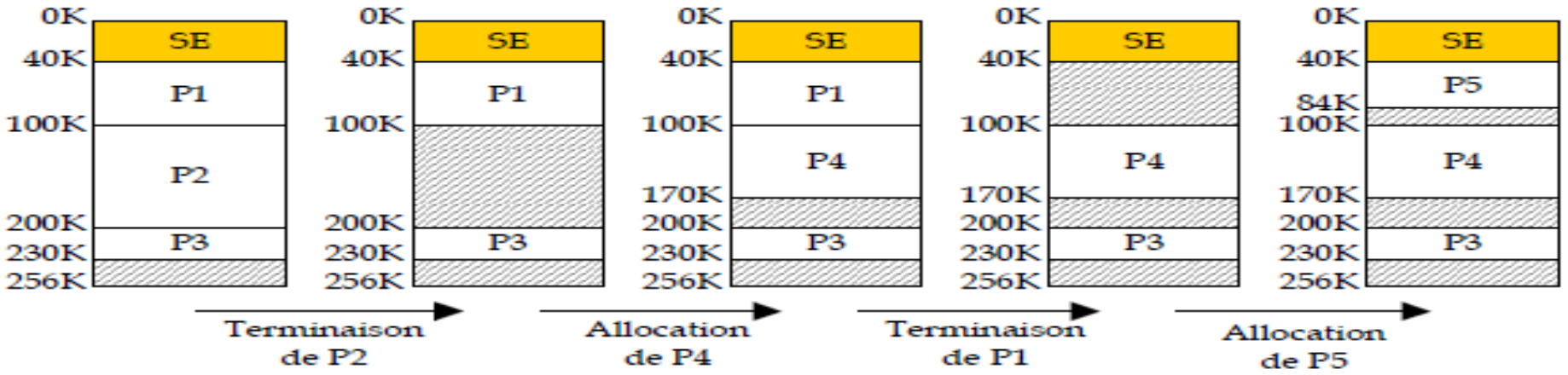
Dr M.CHENAIT
(mchenait@usthb.dz)



Stratégie d'allocation

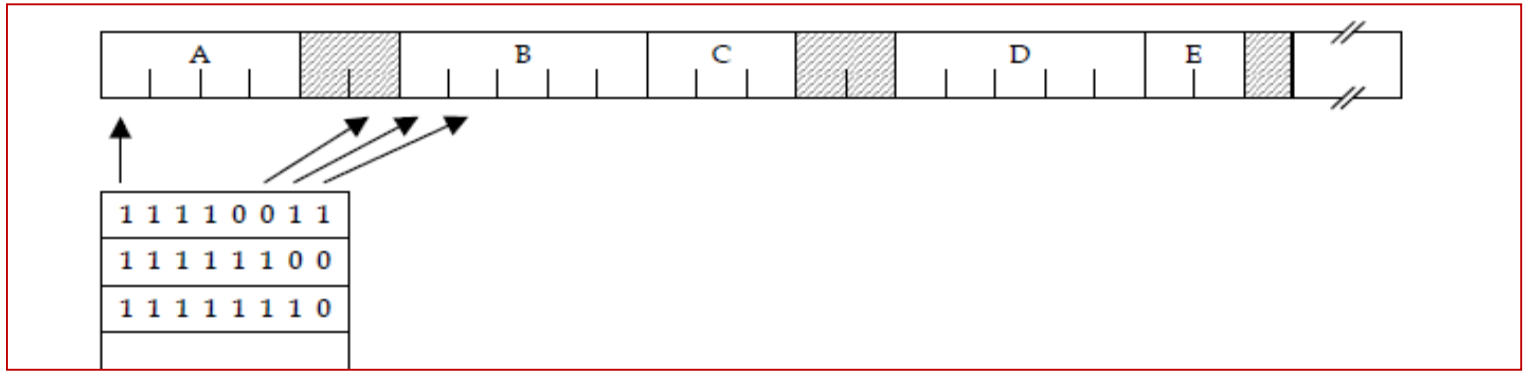
Dr M.CHENAIT
(mchenait@usthb.dz)

- **Partitions multiples variable**

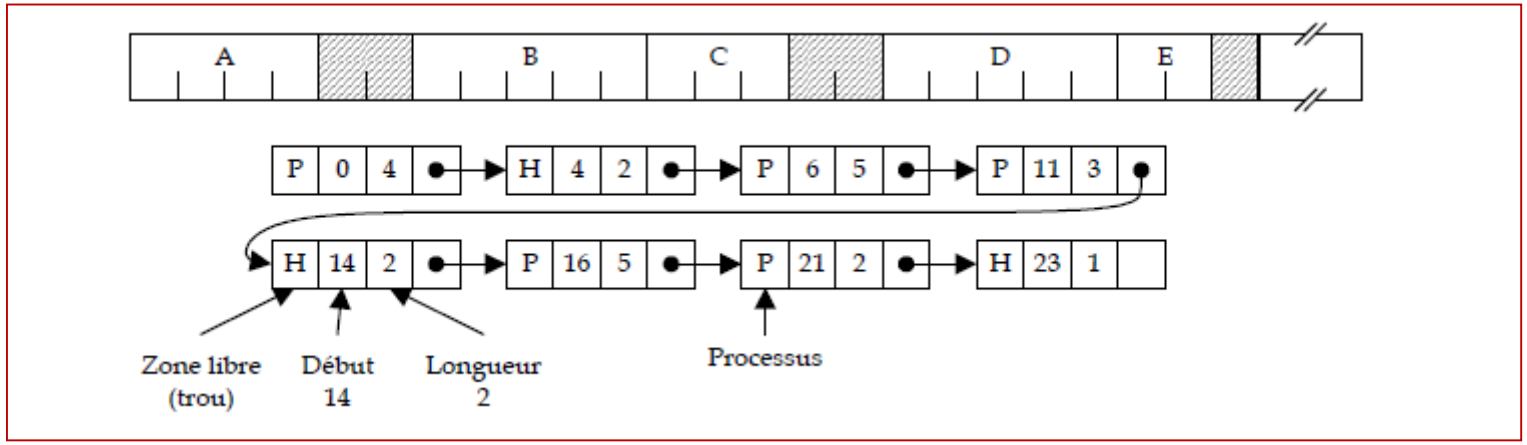


Stratégie d'allocation

Dr M.CHENAIT
(mchenait@usthb.dz)



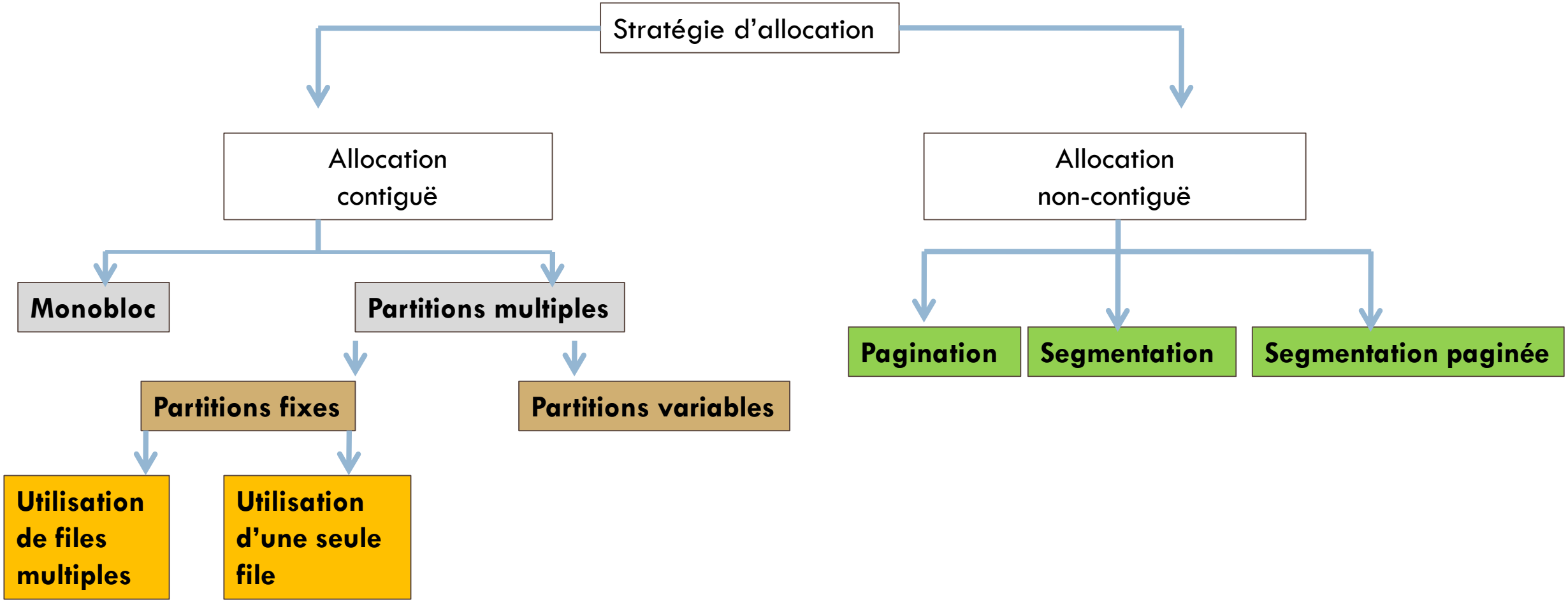
Tables de bits (Bit Maps)



Listes chaînées (Linked List)

Stratégie d'allocation

Dr M.CHENAIT
(mchenait@usthb.dz)



Stratégie d'allocation

Dr M.CHENAIT
(mchenait@usthb.dz)

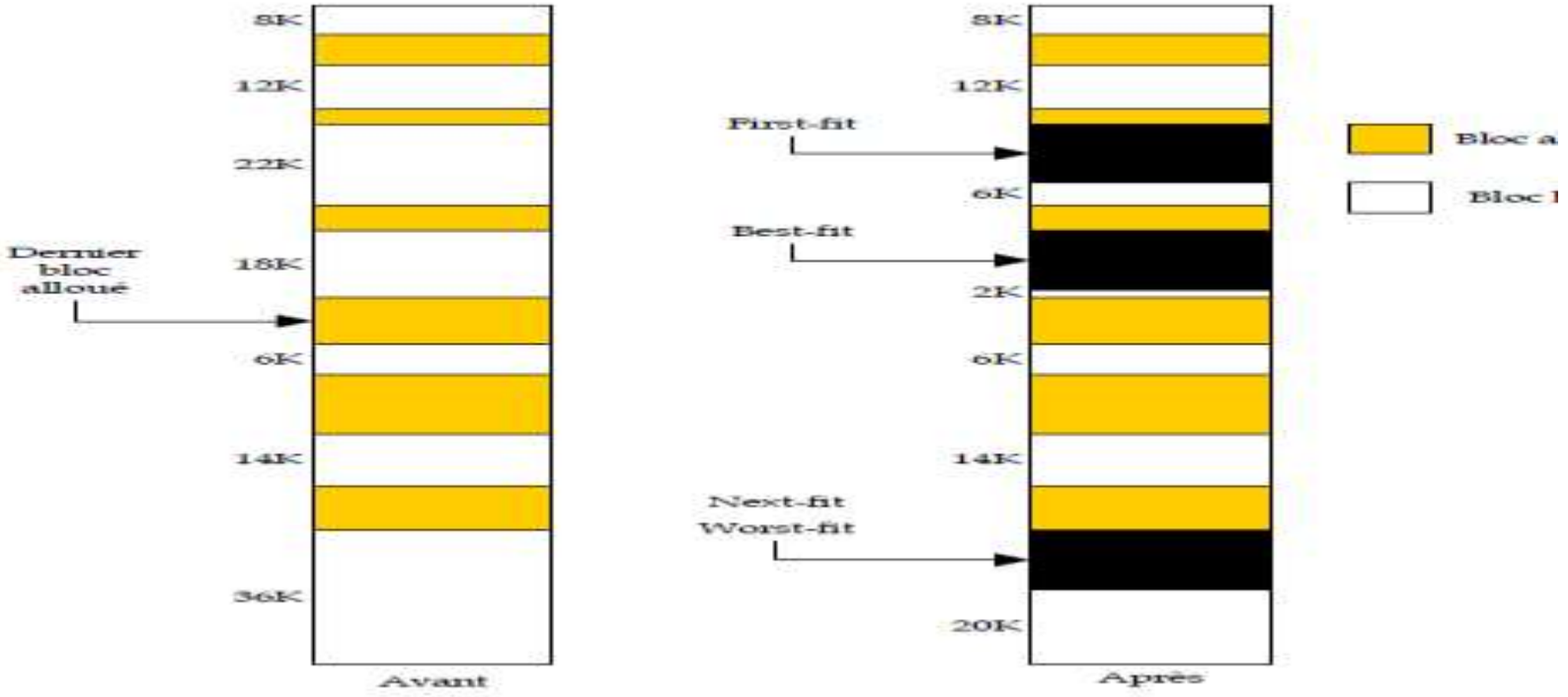
- **Partitions variables- algorithmes de placement-**

_**First-fit** : Le programme est mis dans le premier bloc qui convient à partir du début de la mémoire.
_ **Best-fit** : Le programme est mis dans le bloc de mémoire le plus petit dont la taille est suffisamment grande pour l'espace requis.
– **Worst-fit** : Le programme est mis dans le bloc de mémoire le plus grand pour que la zone libre restante soit la plus grande possible.

Stratégie d'allocation

Dr M.CHENAIT
(mchenait@usthb.dz)

- **Partitions variable- algorithmes de placement-**



Stratégie d'allocation

Dr M.CHENAIT
(mchenait@usthb.dz)

- **Exercice sur les algorithmes de placement**

Exercice 1 de Série n5. (voir corrigé)

Stratégie d'allocation

Dr M.CHENAIT
(mchenait@usthb.dz)

Autres notions

- Fragmentation interne (pb)

- La différence d'espace avant et après allocation dans la partition est appelée fragmentation interne.

- Fragmentation externe (pb)

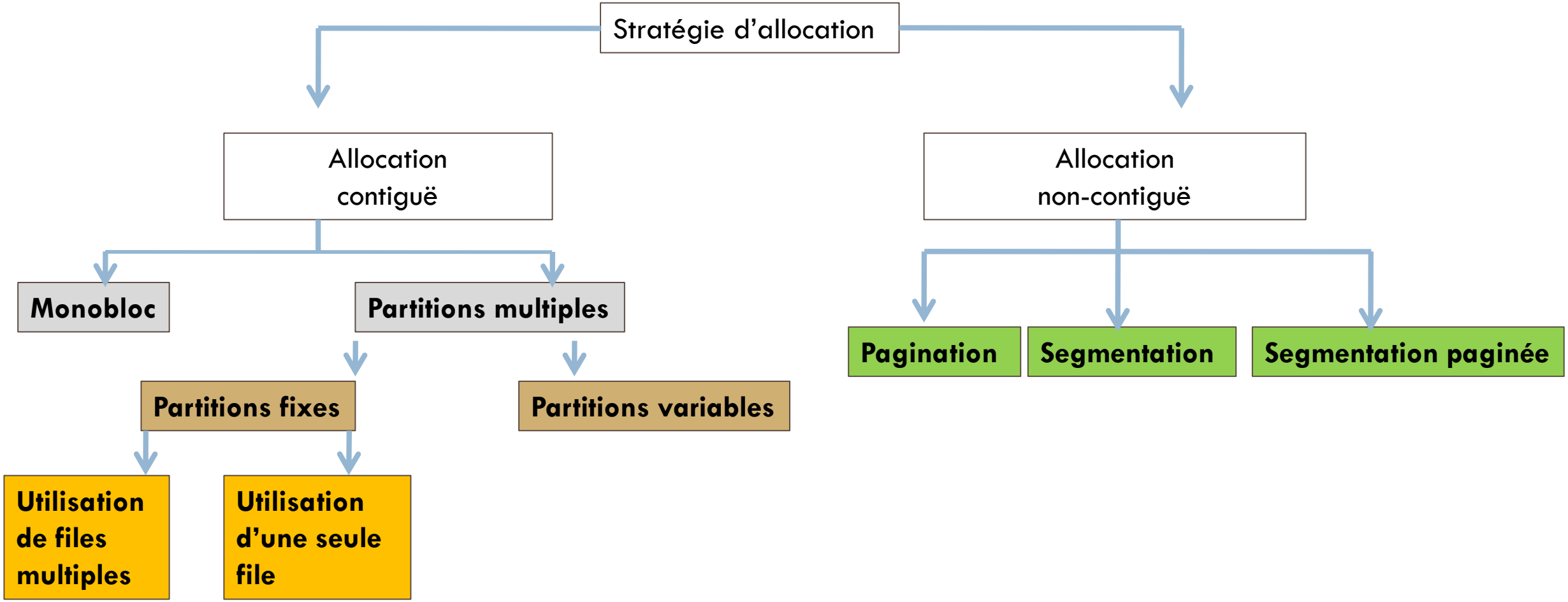
La fragmentation externe se présente quand il existe un espace mémoire total suffisant pour satisfaire une requête, mais il n'est pas contigu ; la mémoire est fragmentée en un grand nombre de petits trous (i.e. blocs libres) où un programme ne peut être chargé dans aucun de ces trous.

- Compactage en MC (comme une défragmentation sur le disque)

Le compactage est une solution pour la fragmentation externe qui permet de regrouper les espaces inutilisés (i.e. les blocs libres) dans une partie de la mémoire (**Ex. début**, milieu).

Stratégie d'allocation

Dr M.CHENAIT
(mchenait@usthb.dz)



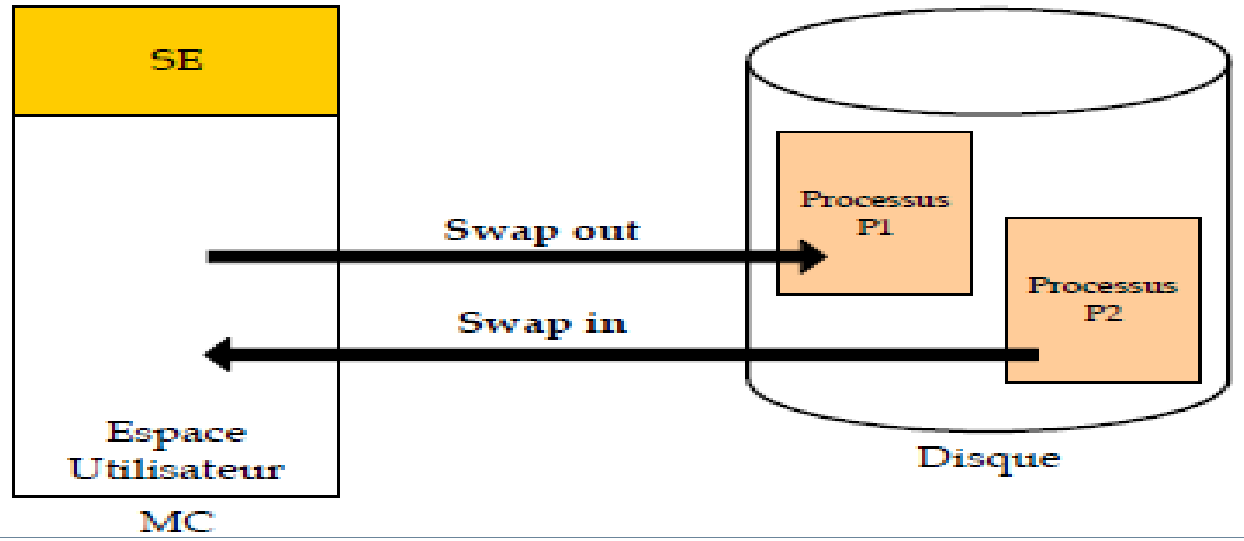
Stratégie d'allocation

Dr M.CHENAIT
(mchenait@usthb.dz)

Autres notions

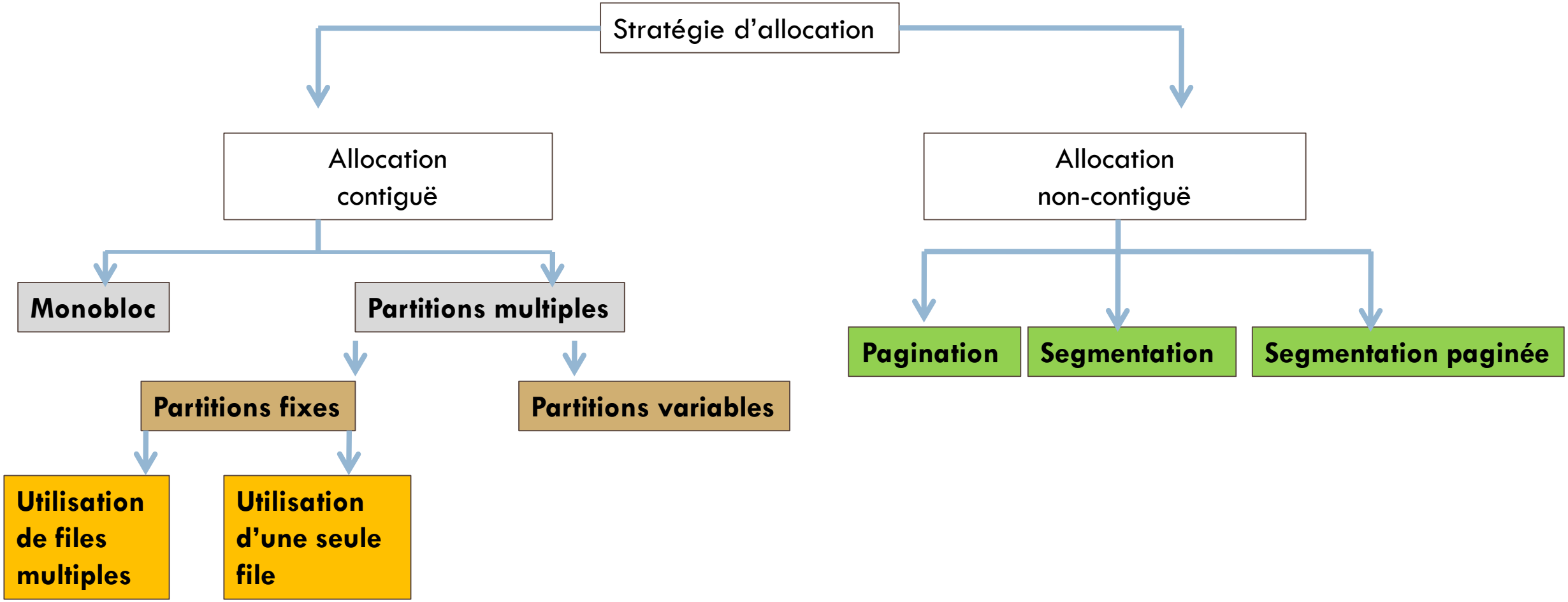
Va-et-vient (Swapping)

Il consiste en le transfert de blocs mémoire de la mémoire secondaire à la mémoire principale ou viceversa
(Swapping).



Stratégie d'allocation

Dr M.CHENAIT
(mchenait@usthb.dz)



Stratégie d'allocation (non- contiguë)

Dr M.CHENAIT
(mchenait@usthb.dz)

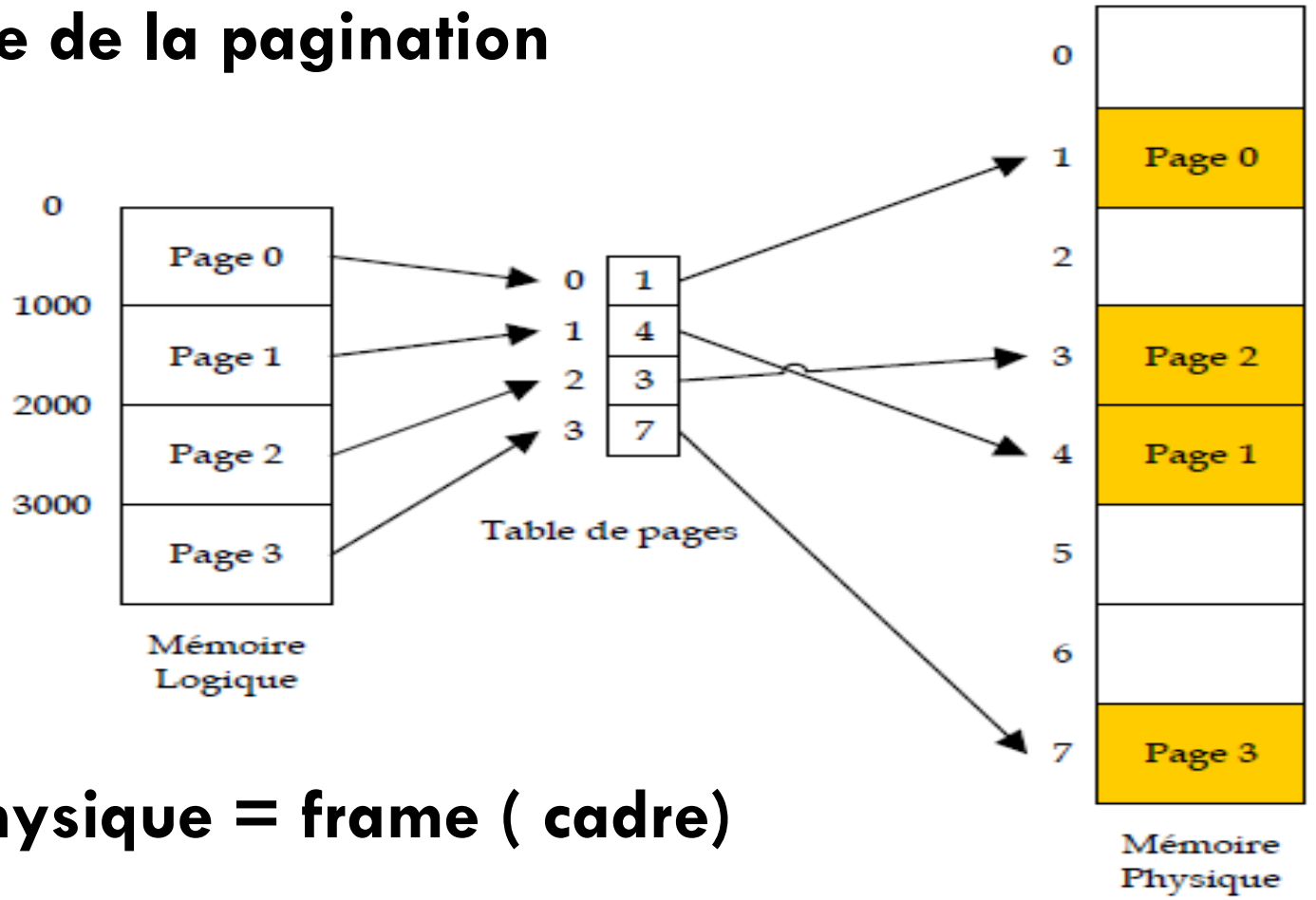
- **Principe de la pagination**

l'espace d'adressage du programme est découpé en petits blocs de **même taille** appelés **pages. L'espace de la mémoire physique est lui-même** découpé en blocs de **taille fixe appelés cases ou cadres de page (Frame Page) ; ce** qui facilitera la **correspondance** d'une page à un frame.

Stratégie d'allocation (non-contiguë)

Dr M.CHENAIT
(mchenait@usthb.dz)

- **Principe de la pagination**

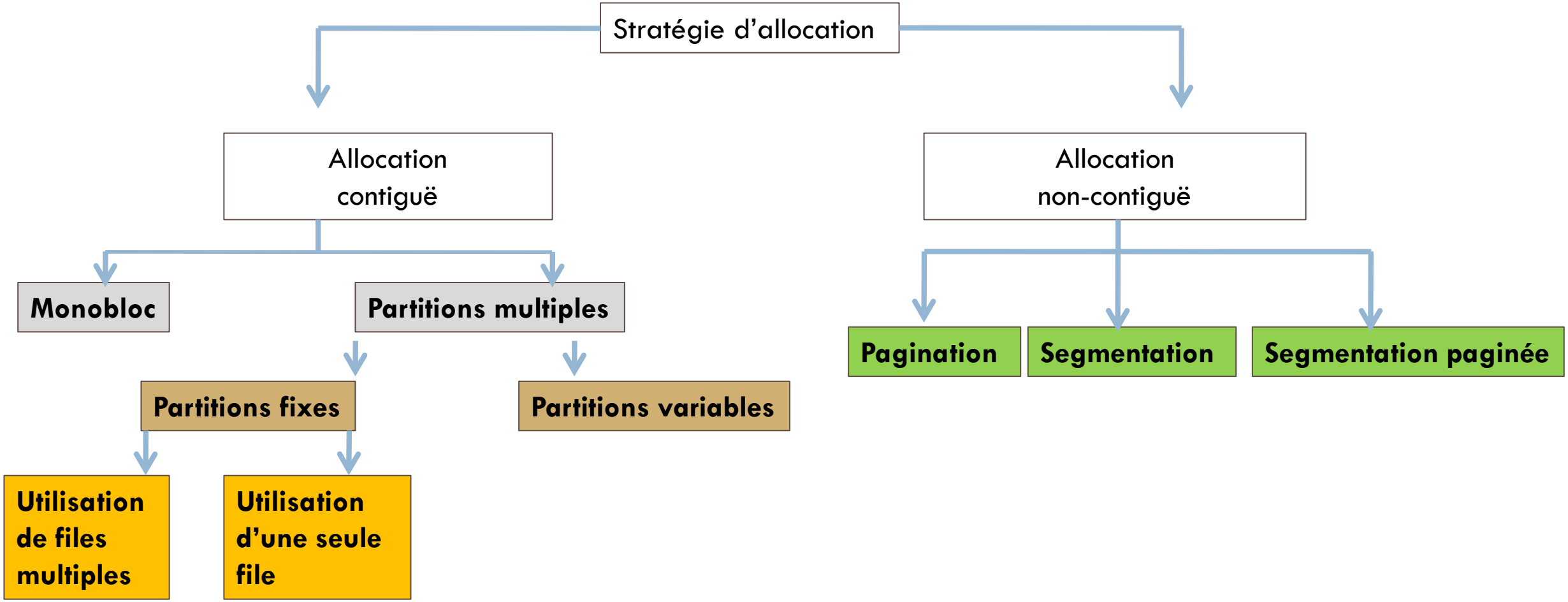


- **Page physique = frame (cadre)**

Swap

Stratégie d'allocation

Dr M.CHENAIT
(mchenait@usthb.dz)

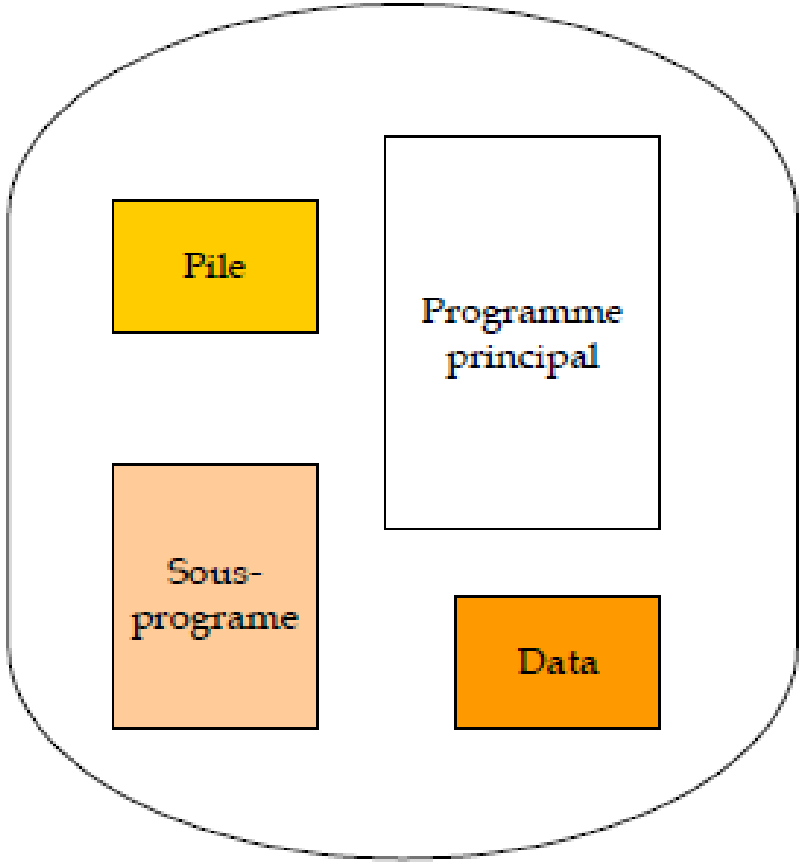


Stratégie d'allocation (non- contiguë)

Dr M.CHENAIT
(mchenait@usthb.dz)

- **Principe de la segmentation**

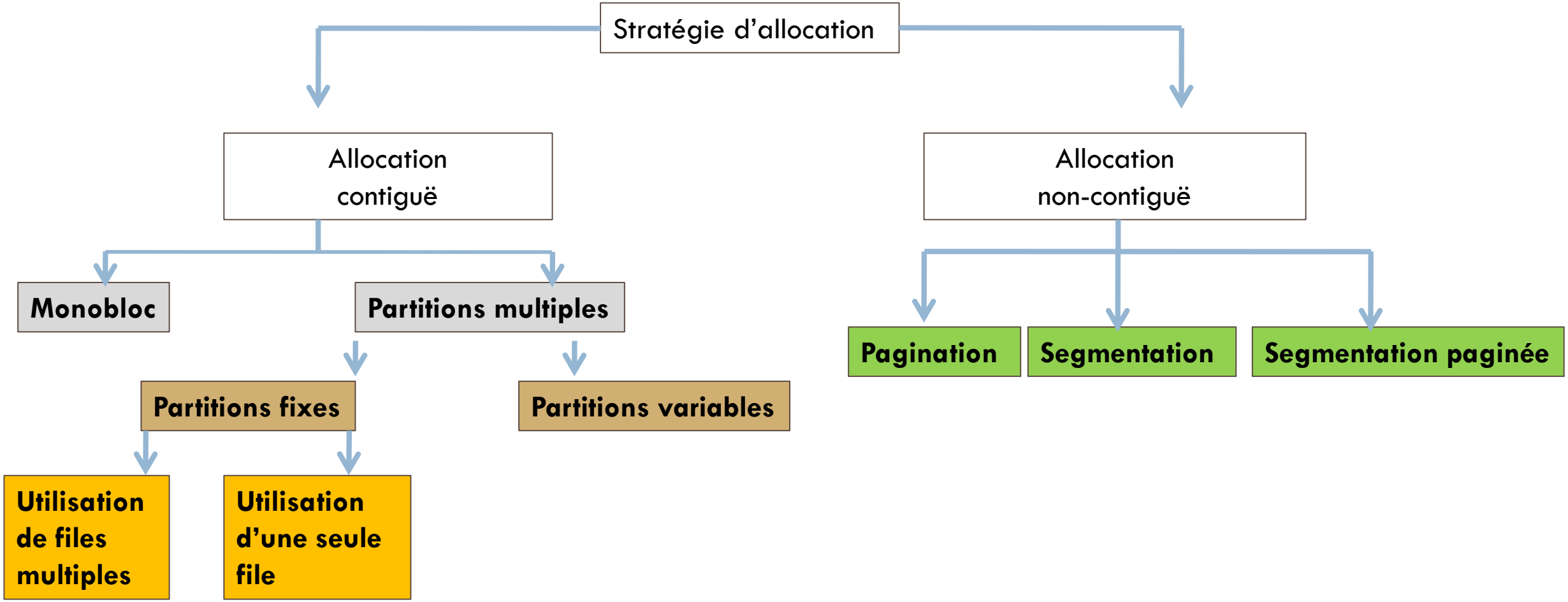
Dans une mémoire segmentée, chaque unité logique d'un programme usager est stockée dans un bloc mémoire, appelé « segment » à l'intérieur duquel les adresses sont relatives au début du segment. Ces segments sont de tailles **différentes**. Un programme sera donc constitué d'un ensemble de segments de code et de données, pouvant être dispersés en MC.



Vue de l'utilisateur d'un programme

Stratégie d'allocation

Dr M.CHENAIT
(mchenait@usthb.dz)

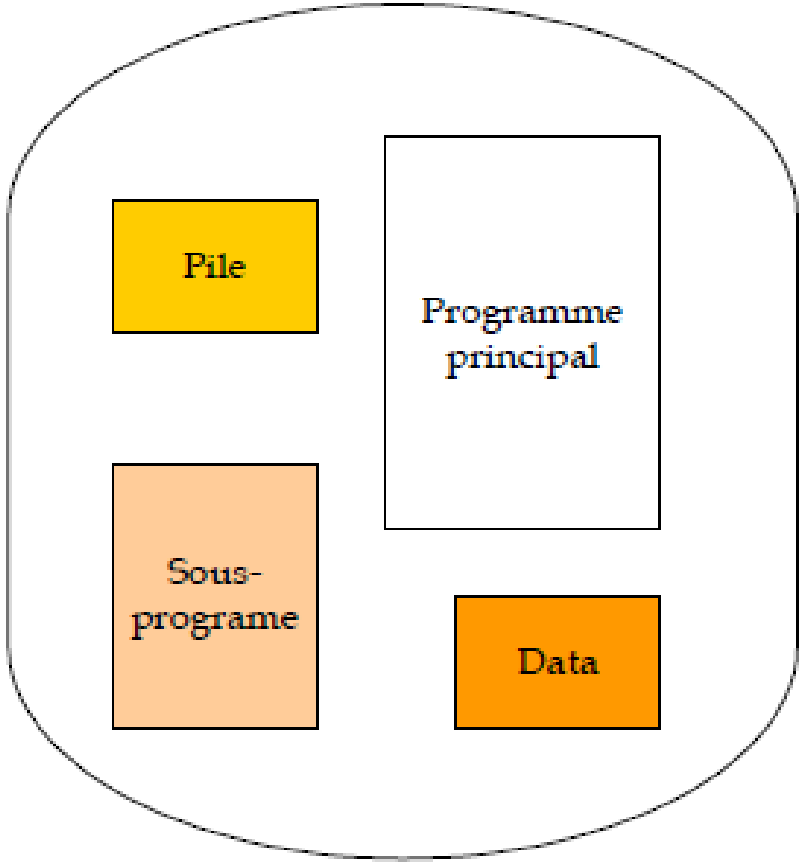


Stratégie d'allocation (non- contiguë)

Dr M.CHENAIT
(mchenait@usthb.dz)

- **Principe de la segmentation paginée**

La taille d'un segment peut être importante, d'où un temps de chargement long qui peut en résulter. La pagination des segments peut être une solution.



Vue de l'utilisateur d'un programme

Stratégie d'allocation

Dr M.CHENAIT
(mchenait@usthb.dz)

- **Exercice sur le calcul d'adresses dans la cas de segmentation**

Exercice 2 de Série n5. (voir corrigé)

La mémoire virtuelle

Dr M.CHENAIT
(mchenait@usthb.dz)

Principe de la MV

•La **mémoire virtuelle (Virtual Memory) est une technique qui permet l'exécution de** processus pouvant ne pas être dans sa totalité en MC.

• Ceci rend possible l'exécution de processus dont la taille est beaucoup plus grande que la taille de la mémoire physique.

• Un processus est constitué de morceaux (pages ou segments) ne nécessitant pas d'être en MC durant l'exécution.

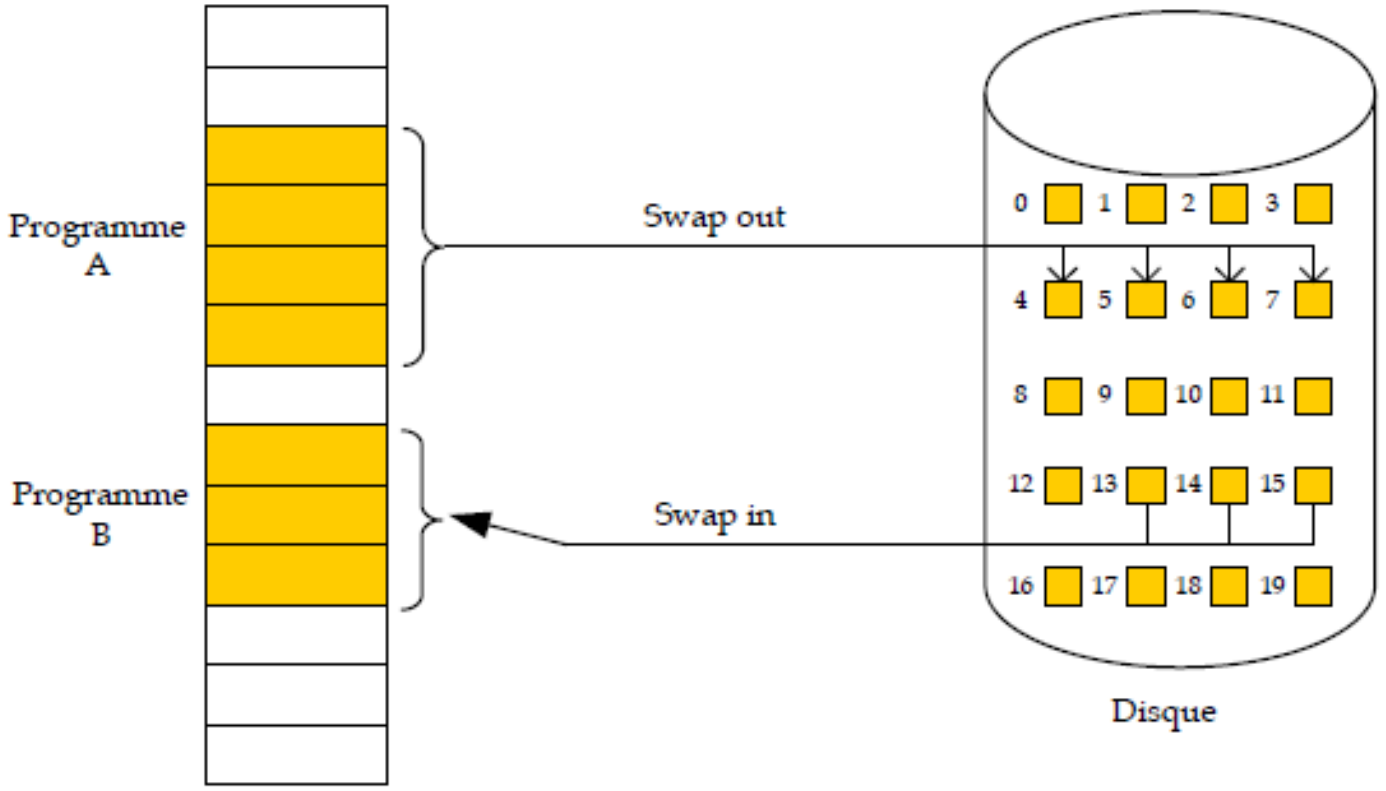
• Pour qu'un programme soit exécuté, seulement les morceaux qui sont en exécution ont besoin d'être en MC.

• Les autres morceaux peuvent être sur mémoire secondaire (**Ex. disque en général**), **prêts à être amenés en mémoire centrale sur demande.**

Dr M.CHENAIT
(mchenait@usthb.dz)

La mémoire virtuelle

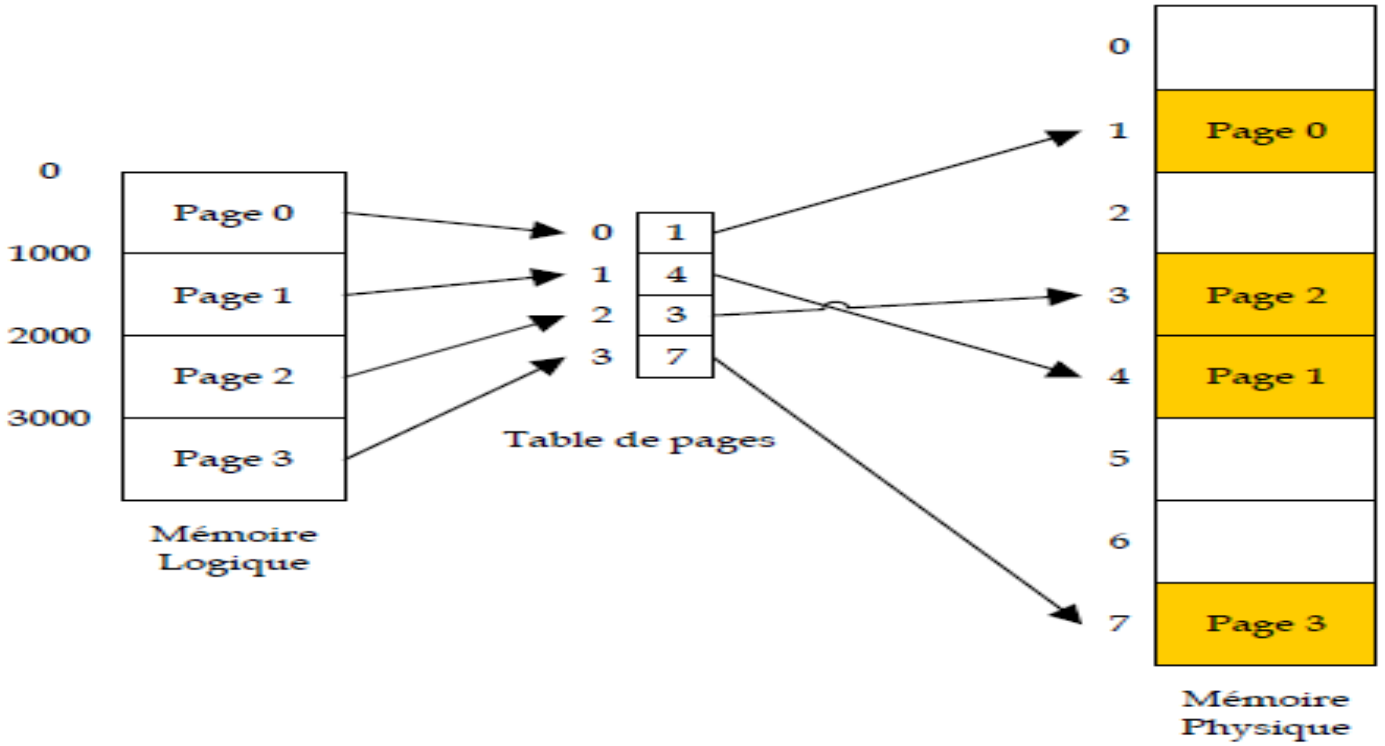
- **Principe de la MV**



Dr M.CHENAIT
(mchenait@usthb.dz)

La mémoire virtuelle

- **Algorithmes de remplacement**
- **Définition d'une page victime**



La mémoire virtuelle

Algorithmes de remplacement

Donne la chaine de référence ds ce cas

Exemple

Soit une mémoire paginée sont la taille d'une page est de 100 octets. L'exécution d'un processus fait référence aux adresses suivantes :

100, 432, 101, 612, 102, 103, 101, 611, 102, 103, 104, 610, 102, 103, 104, 101, 609, 102, 105

Ce

1, 4, 1, 6, 1, 1, 1, 6, 1, 1, 1, 6, 1, 1, 1, 1, 6, 1, 1= chaine de référence = page visitées par le processus lors d'une execution
Chaine réduite = 1,4,1,6,1,6,1,6,1

Dr M.CHENAIT
(mchenait@usthb.dz)

La mémoire virtuelle

- **Algorithmes de remplacement**

- **1. FIFO**
- **2. Optimal**
- **3 LRU (Least Recently Used)**
- **4. Seconde chance**

Dr M.CHENAIT
(mchenait@usthb.dz)

La mémoire virtuelle

- **FIFO:**

	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
Frame 1	7	7	7	2	2	2	2	4	4	4	0	0	0	0	0	0	0	7	7	7
Frame 2		0	0	0	0	3	3	3	2	2	2	2	2	1	1	1	1	1	0	0
Frame 3			1	1	1	1	0	0	0	3	3	3	3	3	2	2	2	2	2	1
Défaut de page	D	D	D	D		D	D	D	D	D	D			D	D			D	D	D
Taux de défaut de page							(15/20) * 100 = 75%													

La mémoire virtuelle

- **FIFO - Anomalie de Baladv-**

	1	2	3	4	1	2	5	1	2	3	4	5
Frame 1	1	1	1	4	4	4	5	5	5	5	5	5
Frame 2		2	2	2	1	1	1	1	1	3	3	3
Frame 3			3	3	3	2	2	2	2	2	4	4
Défaut de page	D	D	D	D	D	D	D			D	D	
Taux de défaut de page							(9/12) * 100 = 75%					

	1	2	3	4	1	2	5	1	2	3	4	5
Frame 1	1	1	1	1	1	1	5	5	5	5	4	4
Frame 2		2	2	2	2	2	2	1	1	1	1	5
Frame 3			3	3	3	3	3	3	2	2	2	2
Frame 4				4	4	4	4	4	4	3	3	3
Défaut de page	D	D	D	D			D	D	D	D	D	D
Taux de défaut de page							(10/12) * 100 = 83,33%					

Avec l'algorithme LRU, **la page la moins récemment utilisée est retirée**. Cette stratégie utilise le principe de la **localité temporelle** selon lequel les pages récemment utilisées sont celles qui seront

Dr M.CHENAIT
(mchenait@usthb.dz)

La mémoire virtuelle

- **Optimal (Algorithme de Belady)**

	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
Frame 1	7	7	7	2	2	2	2	2	2	2	2	2	2	2	2	2	2	7	7	7
Frame 2		0	0	0	0	0	0	4	4	4	0	0	0	0	0	0	0	0	0	0
Frame 3			1	1	1	3	3	3	3	3	3	3	3	1	1	1	1	1	1	1
Défaut de page	D	D	D	D		D		D			D			D				D		
Taux de défaut de page							(9/20) * 100 = 45%													

Avec l'algorithme LRU, **la page la moins récemment utilisée est retirée**. Cette stratégie utilise le principe de la **localité temporelle** selon lequel les pages récemment utilisées sont celles qui seront

Dr M.CHENAIT
(mchenait@usthb.dz)

La mémoire virtuelle

- **LRU (Least Recently Used) moins récemment utilisée**

	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
Frame 1	7	7	7	2	2	2	2	4	4	4	0	0	0	1	1	1	1	1	1	1
Frame 2		0	0	0	0	0	0	0	0	3	3	3	3	3	3	0	0	0	0	0
Frame 3			1	1	1	3	3	3	2	2	2	2	2	2	2	2	2	7	7	7
Défaut de page	D	D	D	D		D		D	D	D	D			D		D		D		
Taux de défaut de page								(12/20) * 100 = 60%												

Avec l'algorithme LRU, **la page la moins récemment utilisée est retirée**. Cette stratégie utilise le principe de la **localité temporelle** selon lequel les pages récemment utilisées sont celles qui seront

Dr M.CHENAIT
(mchenait@usthb.dz)

La mémoire virtuelle

- **Seconde chance // combine les solutions précédant**

	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
Frame 1	7	7	7	2	2	2	2	4	4	4	4	3	3	3	3	0	0	0	0	0
Frame 2		0	0	0	0	0	0	0	2	2	2	2	2	1	1	1	1	7	7	7
Frame 3			1	1	1	3	3	3	3	3	0	0	0	0	2	2	2	2	2	1
Défaut de page	D	D	D	D		D		D	D		D	D		D	D	D		D		D
Taux de défaut de page							(14/20) * 100 = 70%													