



Université Yahia Farès de Médéa
Département: GEI
2^{ème} année Informatique (L.M.D)
Option : RSI/SIIA

Gestion de la Mémoire centrale

Partie : 02

A. KHELDOUN

Émail: univ.medeo@gmail.com

Année Universitaire: 2013-2014

Mémoire Virtuelle

- ❑ **Idée**: Exécution d'un processus partiellement chargé en mémoire
- ❑ **But** : Permettre l'exécution de programmes dont la taille excède celle de la mémoire réelle.
- ❑ Il y a deux types d'adresses dans les systèmes implantant la mémoire virtuelle :
 - Celles référencées par les processus (adresses virtuelles ou logiques),
 - Celles de la mémoire physique (adresses physiques ou réelles).
- ❑ L'espace d'adressage virtuel d'un processus = { **adresses virtuelles que le processus peut référencer**}
- ❑ Sa taille maximale dépend de l'organisation de l'espace virtuel (>> à celle de la mémoire physique).

Mémoire Virtuelle

- ❑ D'une manière générale, l'espace d'adressage est structuré en **blocs** de :
 - mêmes tailles (**pages**-> **pagination**) ou
 - tailles différentes (**segments**->**segmentation**).
- ❑ Les deux organisations peuvent être combinées: **segmentation/pagination**
- ❑ Le format de l'adresse virtuelle est :
 - **Pagination**: (numéro de page, déplacement dans la page);
 - **Segmentation**: (numéro de segment, déplacement dans le segment) pour la;
 - **segmentation/pagination**: (numéro de segment, numéro de page, déplacement dans la page).

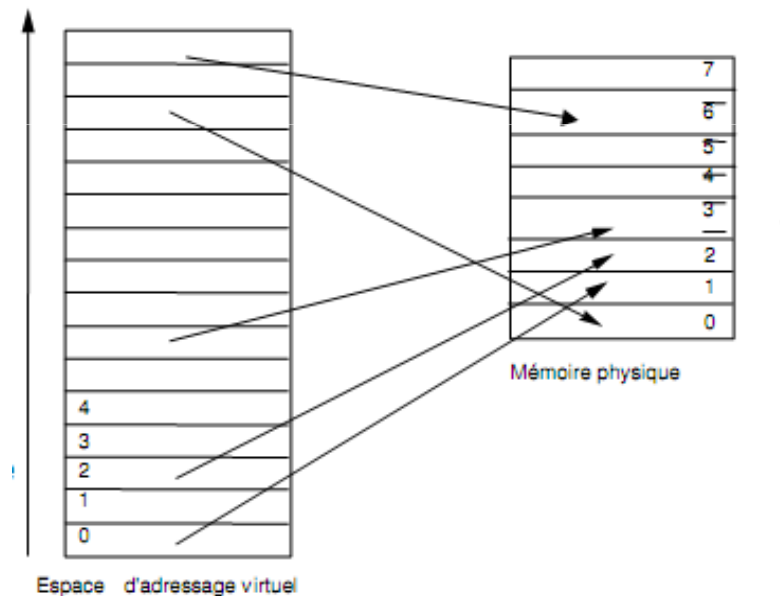
Pagination

- ❑ La mémoire virtuelle et la mémoire physique sont structurées en unités d'allocations (**pages** pour la mémoire virtuelle et **cases** pour la mémoire physique).
- ❑ La taille d'une page est fixe et égale à celle d'une case. Elle varie entre 512 octets et 8 Ko.
- ❑ Il n'y a pas de fragmentation externe car toutes les pages sont de même taille.
- ❑ Par contre, il peut y avoir une fragmentation interne si la dernière page de l'espace d'adressage logique n'est pas pleine.

Pagination

Exemple

- ❑ Soit un programme de 64 Ko sur une machine 32 Ko de mémoire physique.
- ❑ Son espace d'adressage virtuel est composé de 16 pages de 4Ko.
- ❑ La mémoire physique est structurée en 8 cases (cadres) de 4 Ko.



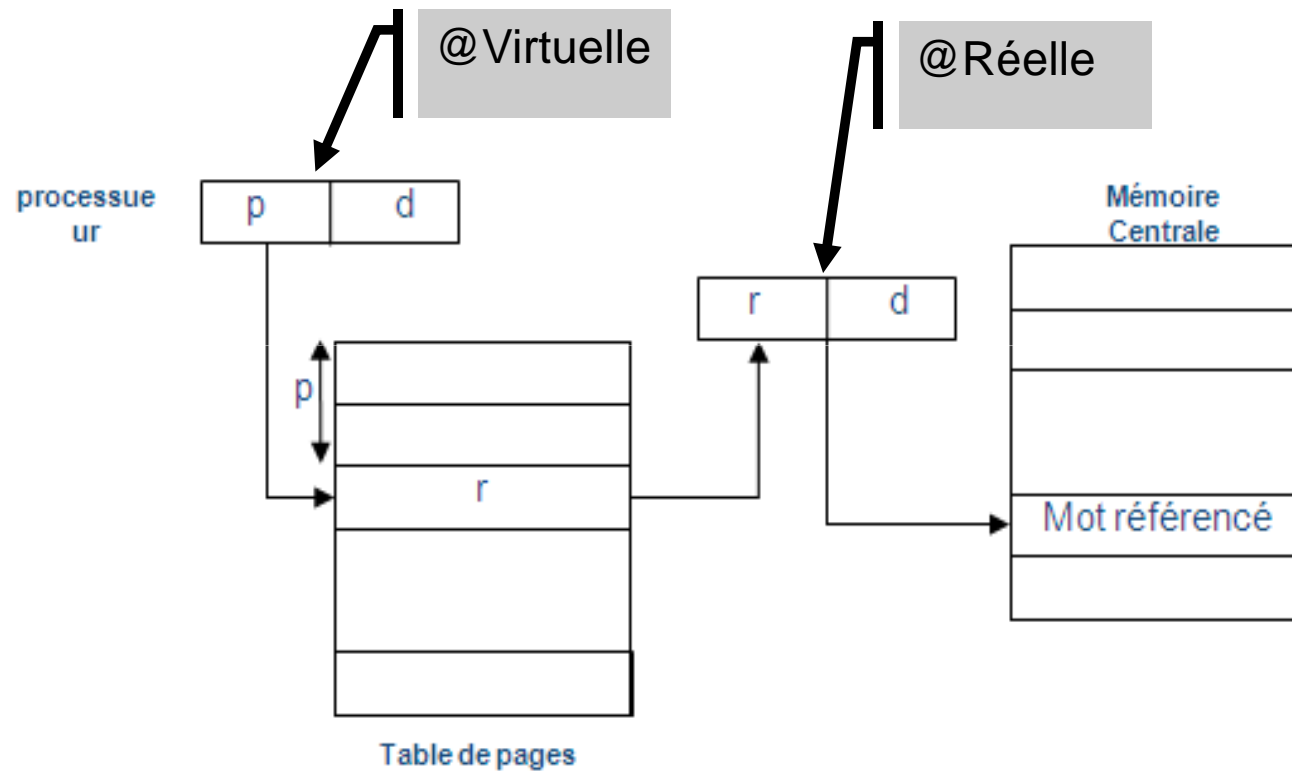
Pagination

Traduction des adresses virtuelles en adresses physiques

- ❑ Adresse virtuelle = (numéro de page, déplacement dans la page).
- ❑ Les adresses virtuelles référencées par l'instruction en cours d'exécution doivent être converties en adresses physiques.
- ❑ La correspondance entre les pages et les cases est mémorisée dans une table appelée **table des pages**. Le nombre d'entrées dans la table est égal au nombre de pages virtuelles.
- ❑ La table des pages d'un processus doit être (en **totalité** ou en **partie**) en mémoire centrale lors de l'exécution du processus. Elle est nécessaire pour la conversion des adresses virtuelles en adresses physiques.
- ❑ Chaque entrée de la table des pages contient le numéro de case (adresse physique) de la page correspondante (et d'autres informations que l'on détaillera ultérieurement).

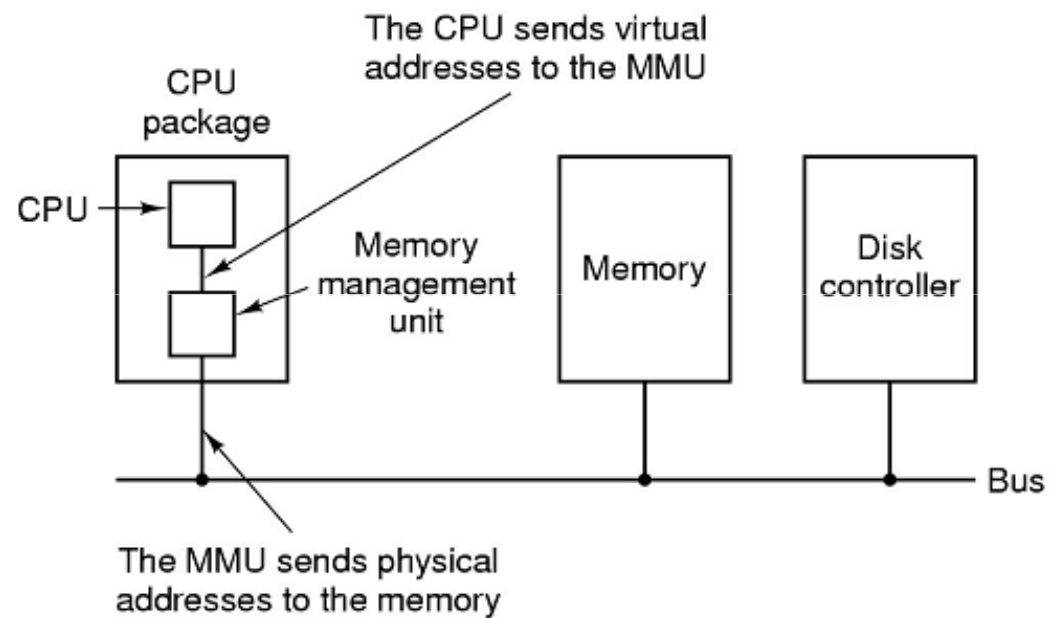
Pagination

Mécanisme de Traduction



Pagination

Support matériel (MMU)

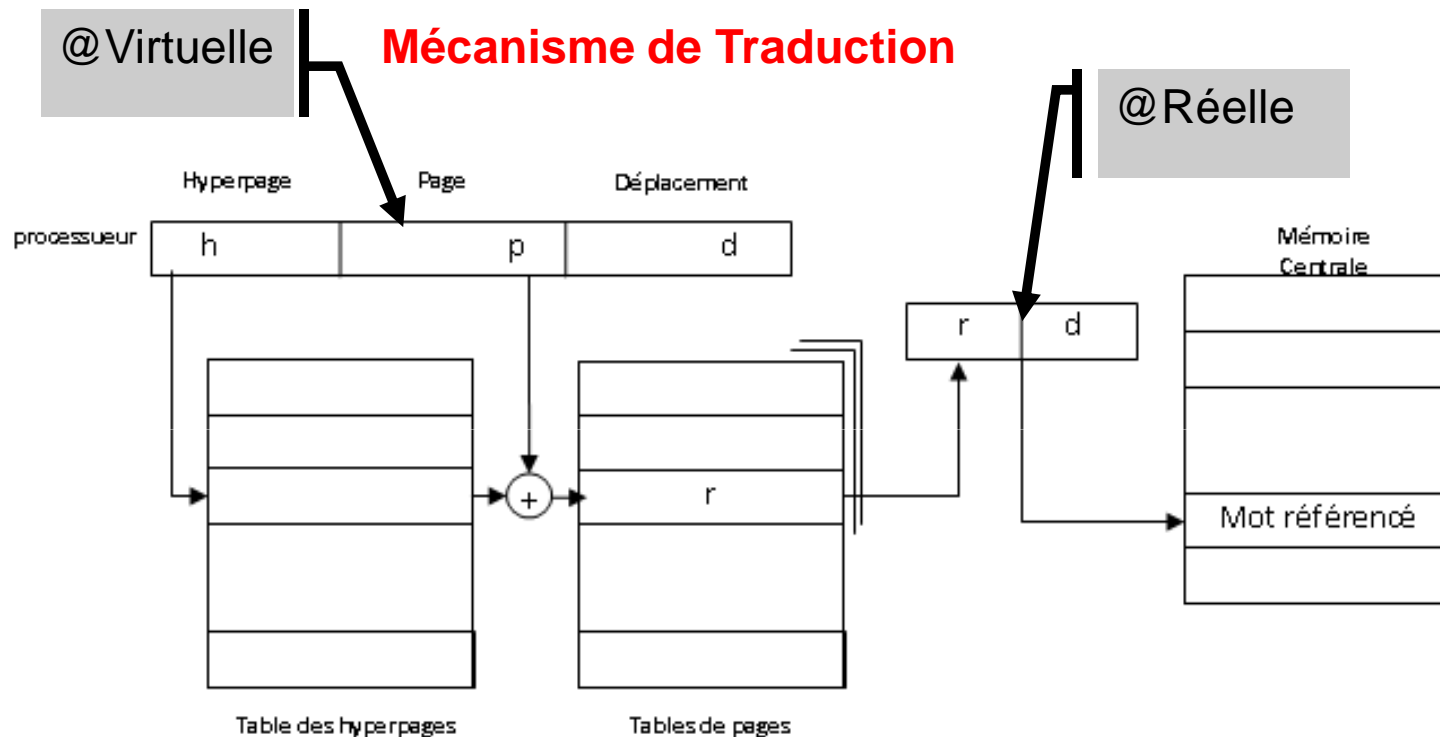


Pagination

Table des pages à deux niveaux

- ❑ La taille de la table des pages peut être très grande : 220 entrées (plus d'1 million) pour un adressage virtuel sur 32 bits et des pages de 4 Ko.
- ❑ Pour éviter d'avoir des tables trop grandes en mémoire, de nombreux ordinateurs utilisent des **tables des pages à plusieurs niveaux**.
- ❑ Par exemple, une table des pages à deux niveaux, pour un adressage sur 32 bits et des pages de 4 Ko, est composée 1024 tables de 1024 entrées. il est ainsi possible de charger que les tables nécessaires.
- ❑ Dans ce cas, une **adresse virtuelle** de 32 bits est composée de **trois champs** :
 - un pointeur sur la table du 1^{er} niveau (10 bits), → **Hyperpage**
 - un pointeur sur une table du 2nd niveau (10 bits) → **Page**
 - un **déplacement** dans la page (12 bits).

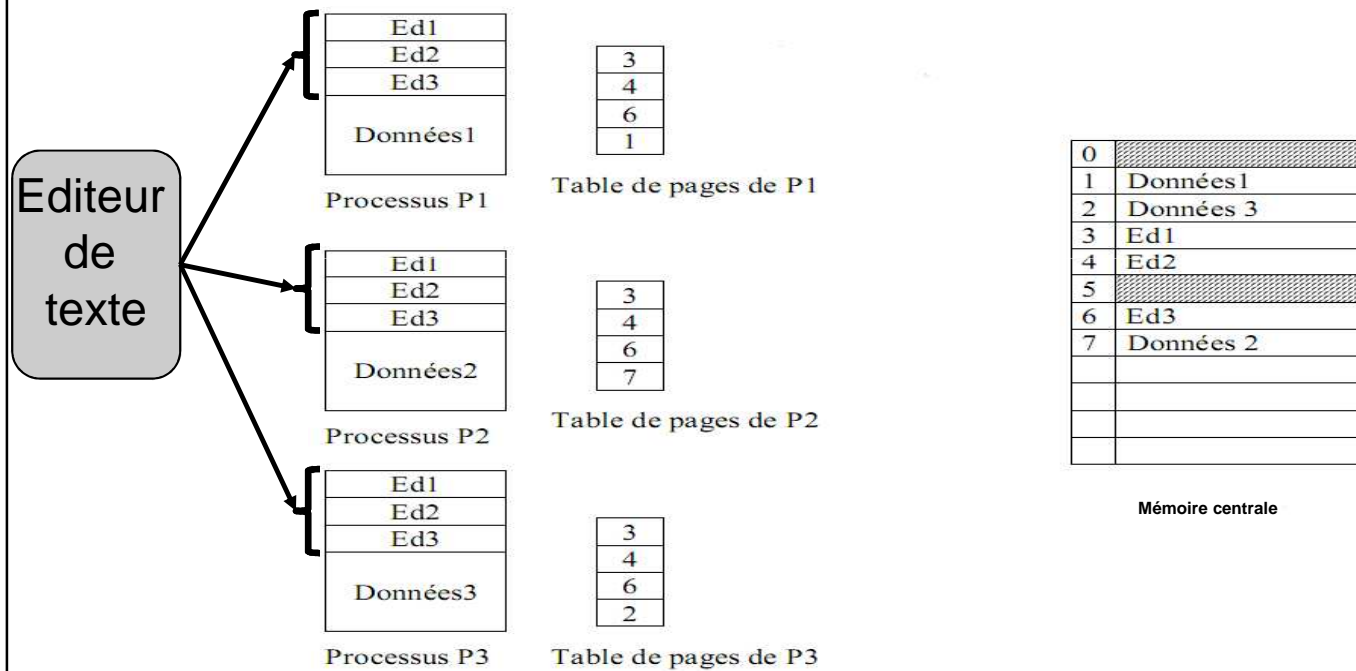
Pagination



Trois accès sont nécessaires pour atteindre un mot en mémoire centrale (un accès à la table des hyperpages, un accès à la table de pages et un accès au mot référencé).

Pagination

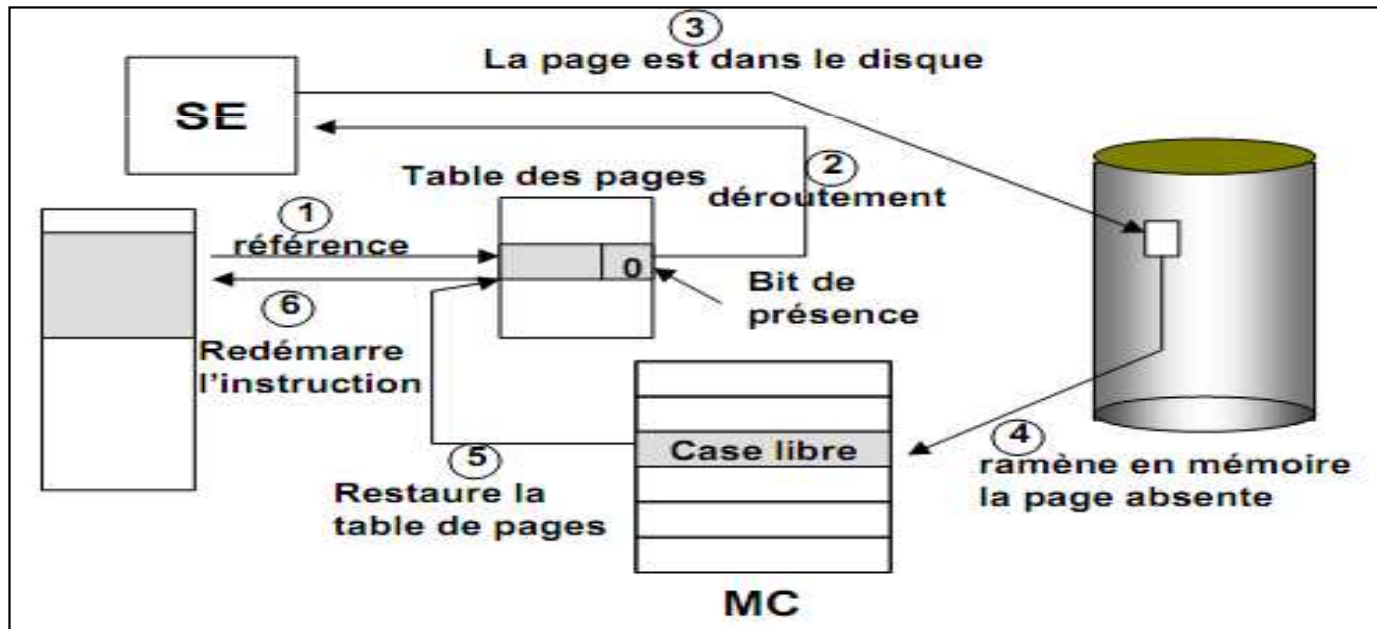
- ❑ Un autre avantage de la pagination est: **Partage de code.**
- ❑ **Exemple**



Pagination

Pagination à la demande

- ❑ **Défaut de page:** Accès à une page **n'est pas en Mémoire Centrale**
- ❑ A la suite d'un défaut de page, le système d'exploitation doit ramener en mémoire la page manquante à partir du disque.



Pagination

Pagination à la demande

- ❑ S'il n'y a pas de cases libres en mémoire, il doit retirer une page de la mémoire pour la remplacer par celle demandée.
- ❑ Si la page à retirer a été modifiée depuis son chargement en mémoire, il faut la réécrire sur le disque.
- ❑ Quelle est la page à retirer de manière à minimiser le nombre de défauts de page ?

→ **Politique de remplacement de pages**

Pagination

Algorithmes de remplacement de pages

A. Algorithme FIFO: remplacer la page la plus **ancienne**

Exemple: MC = 03 cases

Chaîne de référence	→	7	0	1	2	0	3	0	4	2	3
Cadres de page	{	7	7	7	2	2	2	2	4	4	4
			0	0	0	0	3	3	3	2	2
				1	1	1	1	0	0	0	3
Défauts de page	→	X	X	X	X		X	X	X	X	X

Nombre de défauts de pages : 9 (9/10=90%).

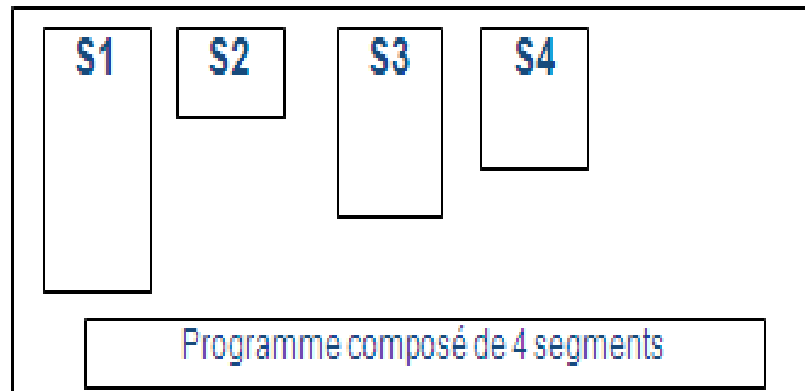
B. Algorithme LRU (*Least Recently Used*): remplacer la page qui est restée inutilisée pendant le plus de temps

Chaîne de référence	→	7	0	1	2	0	3	0	4	2	3
Cases (pages)	{	7	7	7	2	2	2	2	4	4	4
			0	0	0	0	0	0	0	0	3
				1	1	1	3	3	3	2	2
Défauts de page	→	X	X	X	X		X		X		X

Nombre de défauts de pages : 7 (7/10=70%).

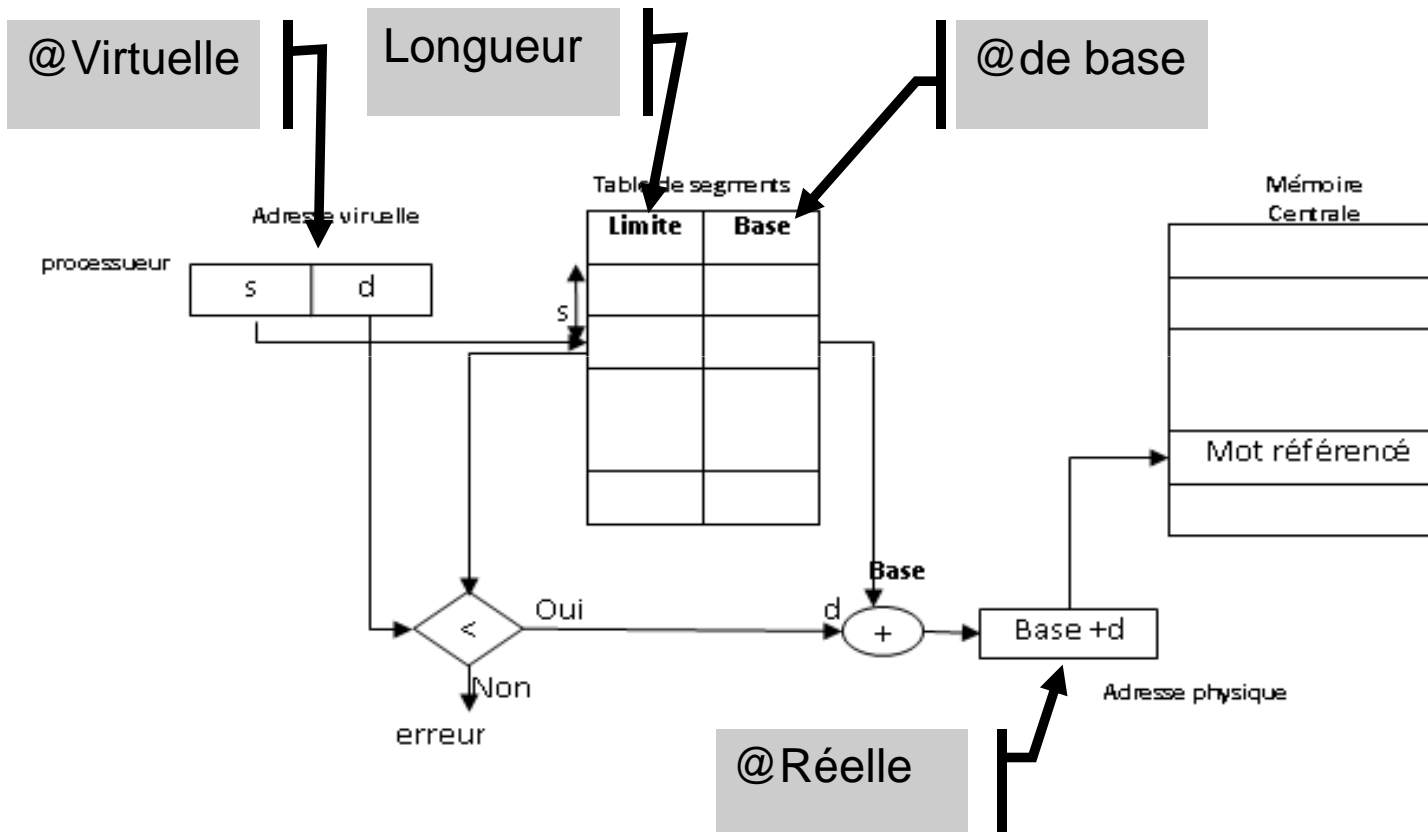
Segmentation

- ❑ Dans un système paginé, l'espace d'adressage virtuel d'un processus est à une dimension.
- ❑ Or en général, un processus est composé d'un ensemble d'unités logiques :
 - Les différents codes : le programme principal, les procédures, les fonctions bibliothèques ;
 - Les données ;
 - Les piles d'exécution.
- ❑ On peut associer à chaque unité logique un espace d'adressage (un segment).



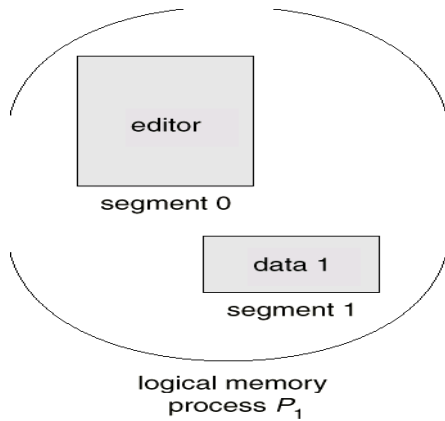
Segmentation

Traduction des adresses virtuelles en dressees physiques



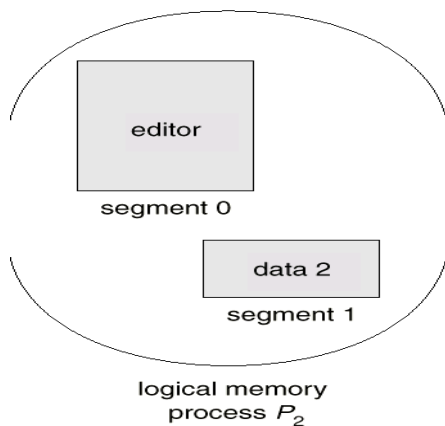
Segmentation

Partage de segment



	limit	base
0	25286	43062
1	4425	68348

segment table
process P_1



	limit	base
0	25286	43062
1	8850	90003

segment table
process P_2

