Cours 07:
Gestion de la Mémoire (Memory Management)

Plan du cours

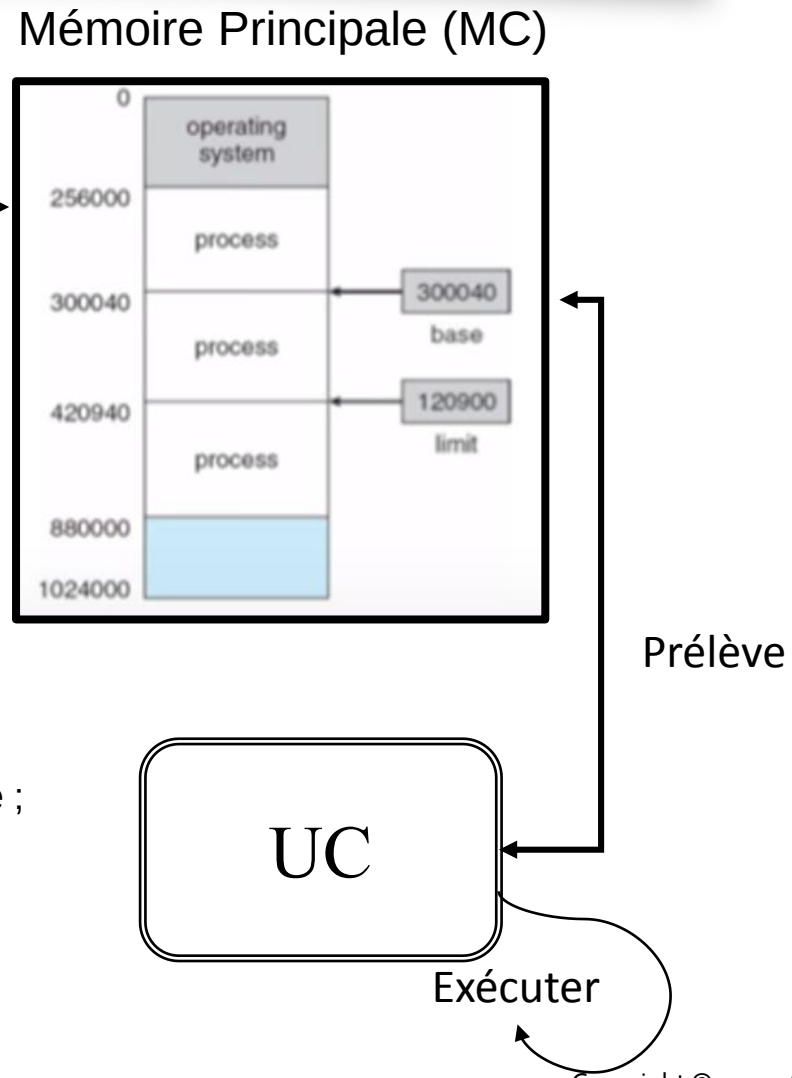
- **Concepts de base**
- **La gestion de mémoire (Pourquoi ?)**
- **La Mémoire virtuelle**
- **La pagination de la mémoire**
- **Détection et traitement d'un défaut de page**

Exécution d'un programme


```
4 void     main()
5    {
6    printf( "Hello, from main\n" );
7    b();
8 }
9
10
11 voidb()
12    {
13    printf( "Hello, from 'b'\n" );
14 }
```

programme système appelé **chargeur**

chargé entièrement ou partiellement

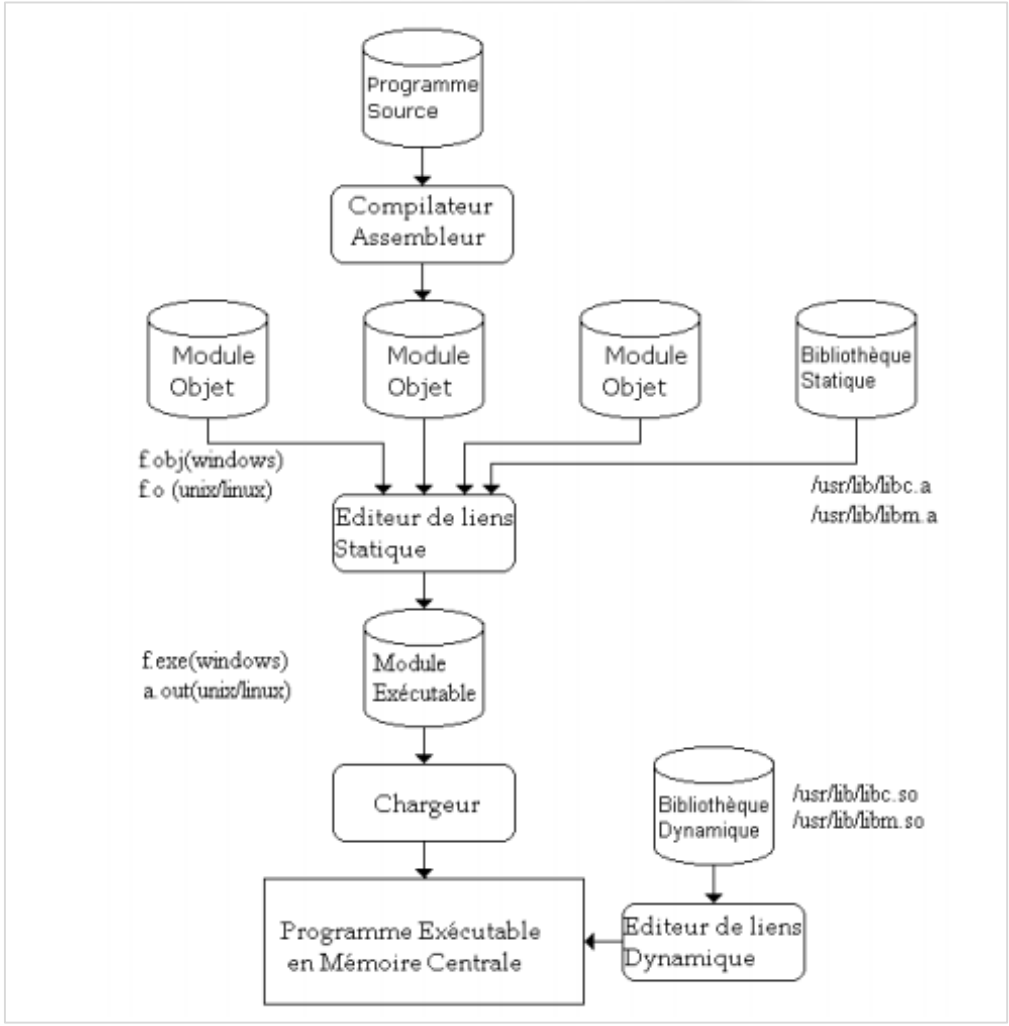


Cycle d'exécution d'une instruction typique

1. Extraction d'une instruction de la mémoire ;
2. L'instruction est décodée ;
3. Elle provoque, éventuellement, l'extraction d'opérandes de la mémoire ;
4. Une fois l'instruction exécutée sur les opérandes,
5. Les résultats peuvent être réenregistrés en mémoire.

Les différentes étapes de l'exécution d'un programme

- Code Source ou Programme (C, C++, Java etc.)
- Module Object
  - Un fichiers non exécutable
  - Non stocké dans des fichiers
- Charger les modules Objet et les Bibl. dans MC
- Après Chargement: Allocation + adressage
- Utiliser Les bibliothèques dynamiques (DLLs)



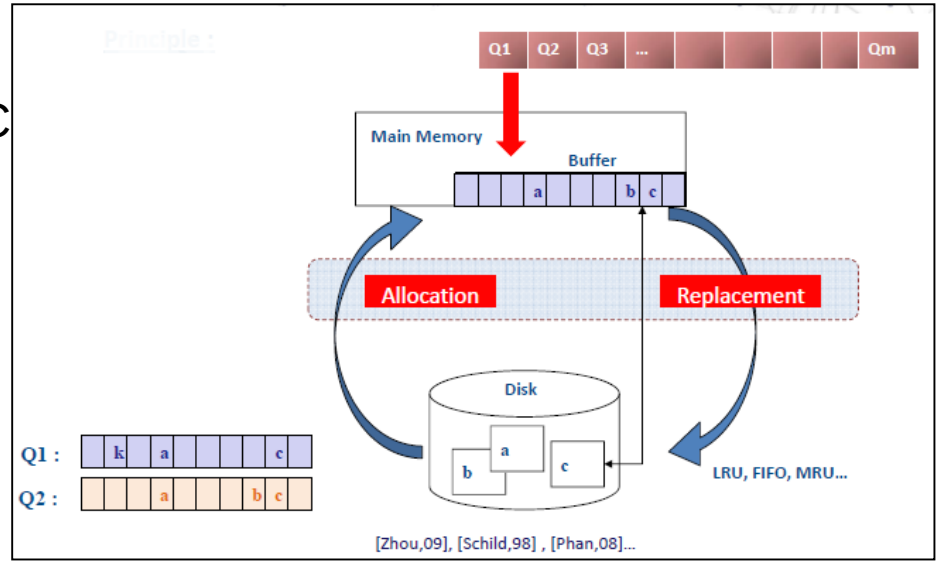
➲ Il existe plusieurs techniques de gestion de la mémoire. Selon l'architecture de la machine.

Débat (5 mn)

- Quelle est la différence entre le codage et le script?

La gestion de mémoire (Pourquoi ?)

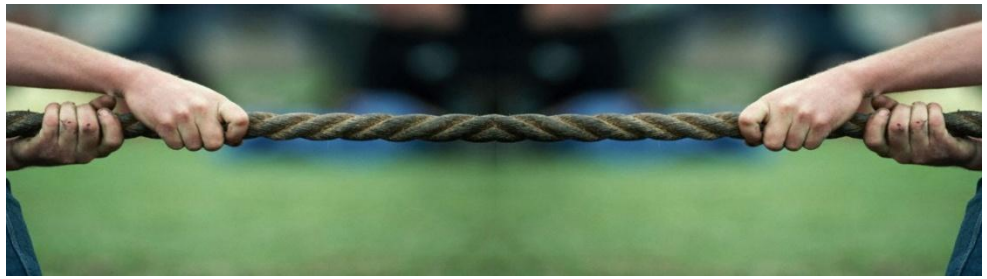
- La mémoire aujourd'hui et devient moins chère
- Mais les applications exigent plus et plus de mémoire
- **Memory Management**: échanger des blocs de données de MS->MC



- Les E/S de mémoire sont lentes par rapport au CPU

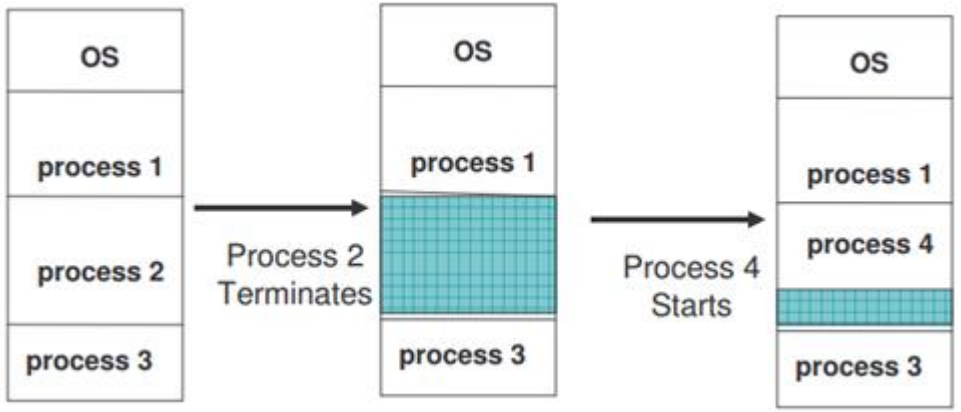
- Le système d'exploitation doit:
  - Maximiser l'efficacité de la CPU
  - Minimiser le temps d'échange

CPU **I/O**



Problèmes

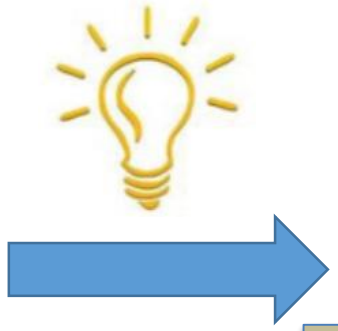
- Problème de capacité de MC ☹
- Cohabitation entre les processus ☹
  - Libérer pour avoir un espace adjacent ?



- Contrôle et le partage des informations entre les processus ☹
  - Concerne les informations critiques
  - Empêcher une modification de certaines informations,

Exigence de gestion de la mémoire

- Réallocation
- Protection
- Contrôle de Partage
- Organisation logique
- Organisation physique

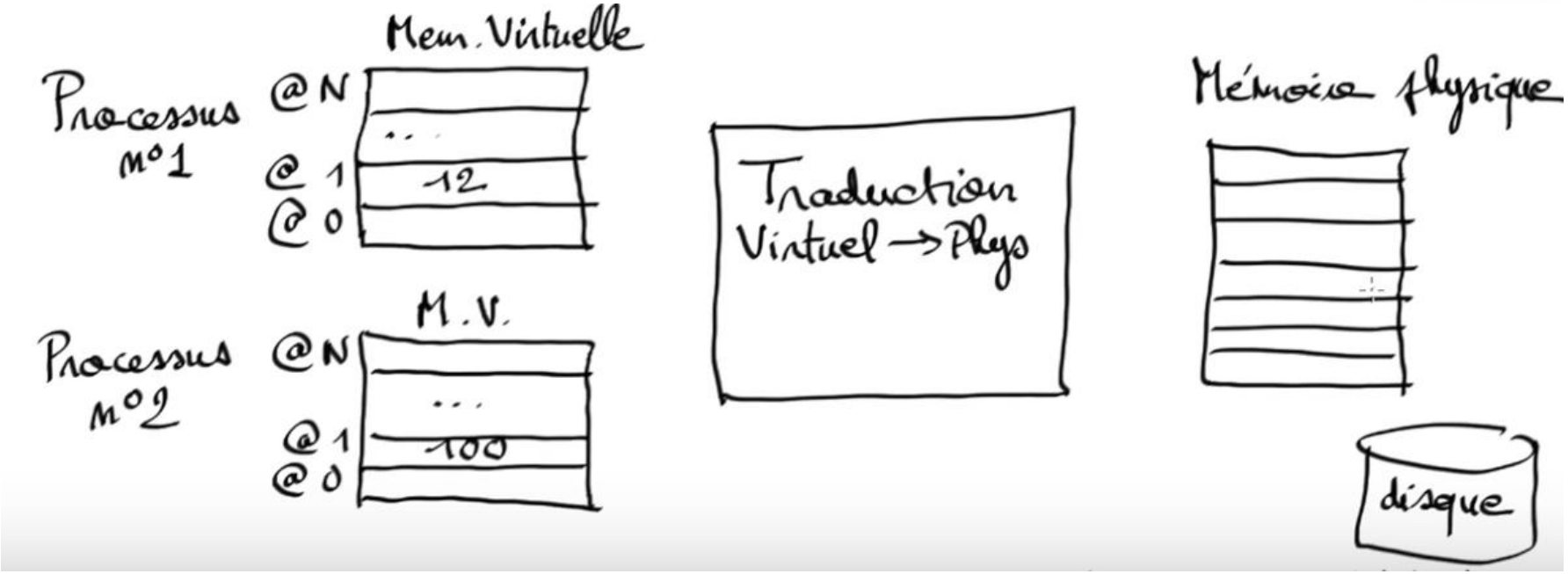


Principe de la solution c'est la
Mémoire virtuelle

➲ **La mémoire virtuelle** est une technique autorisant l'exécution de processus pouvant ne pas être complètement en mémoire.

➲ Avec cette technique, on pourra exécuter des programmes qui peuvent être plus grands que la mémoire physique.

Principe de la solution



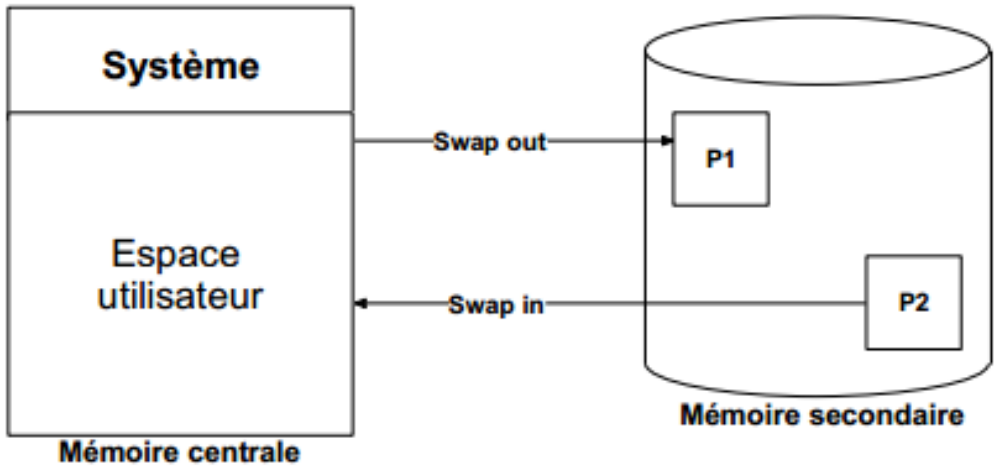
Idées Générales:

- 1 espace de mémoire virtuelle par processus
- @ Virtuelles et @ Physiques
- Echanges transparents avec le disque sur

Swapping

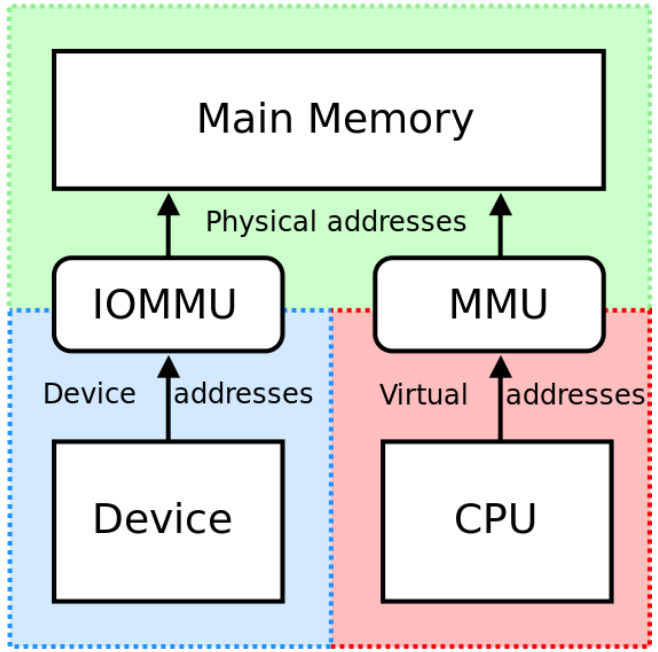
- Le swapping ou échange consiste à déplacer temporairement des processus entre la mémoire centrale et la mémoire auxiliaire (disques).
- L'objectif visé étant soit de libérer de l'espace mémoire pour de nouveaux processus, soit de réduire le taux de multiprogrammation.

- **Un espace swap** (zone d'échange) est donc créé sur le disque.
  - Unix: (/swap).
  - Windows (pagefile.sys).



ADRESSAGE MÉMOIRE

- Une adresse générée par l'UC est dite adresse logique (relative).
- La conversion entre une adresse virtuelle (logique) en une adresse réelle (physique) est réalisée par la MMU (Memory Management Unit) ou unité de gestion mémoire.
- MMU: un dispositif matériel

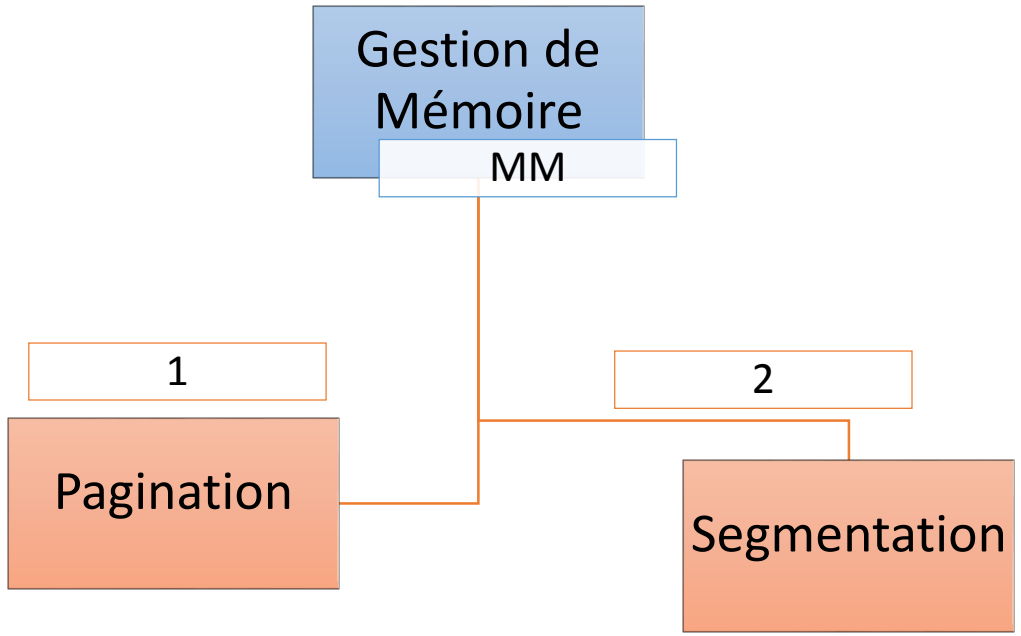


Gestionnaire de la mémoire

- C'est un module système dont la fonction est de gérer la mémoire principale.
- Les principales fonctions d'un gestionnaire de la mémoire sont les suivantes :
  - Allouer / libérer l'espace de la mémoire principale,
  - Charger / décharger(sauvegarder) les programmes des processus,
  - Protéger les programmes et les données des processus,
  - Partager des programmes et/ou des données entre plusieurs processus

Comment implémenter la mémoire virtuelle ?

C'est un terme utilisé pour décrire comment les systèmes d'exploitation gèrent la RAM disponible:



Question ?

☞??

La pagination de la mémoire

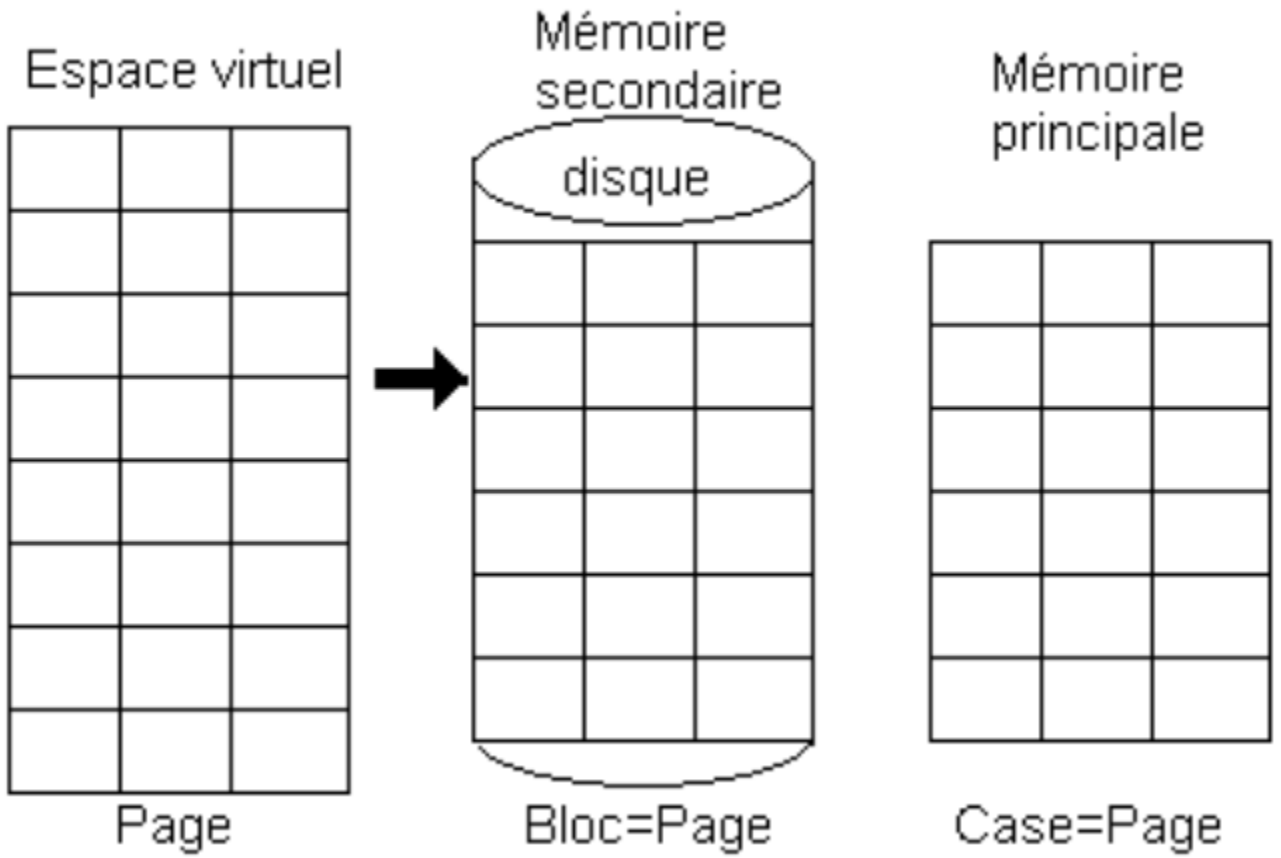
Principe de la pagination

- La pagination consiste à diviser l'espace d'adressage logique en unités ou **blocs appelés pages**.
- La mémoire physique est également découpée en blocs ou unités de même taille que les pages, appelés cases ou cadres de page (pages frames).
- Les pages, des processus (**espaces virtuels**), qui ne sont pas chargées dans des cases de mémoire physique sont stockées en mémoire secondaire (disque).
- La partie du disque (mémoire secondaire) réservée à la pagination est découpée en blocs de même taille que les pages (certains systèmes charge toutes les pages du programme en mémoire secondaire).

Principe de la pagination

Frames = blocs physiques
Pages = blocs logiques

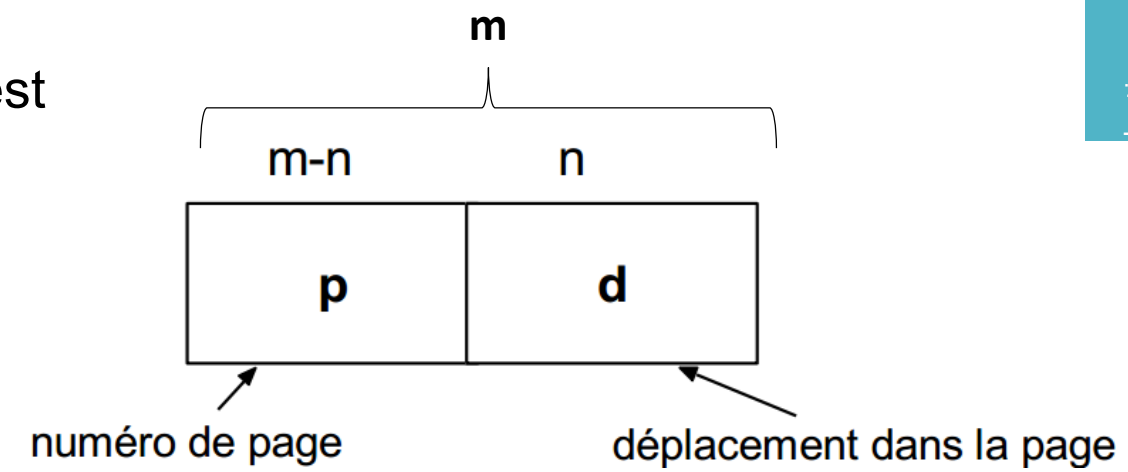
La taille des Frames et des Pages est définie par le matériel (puissance de 2 pour faciliter les calculs)



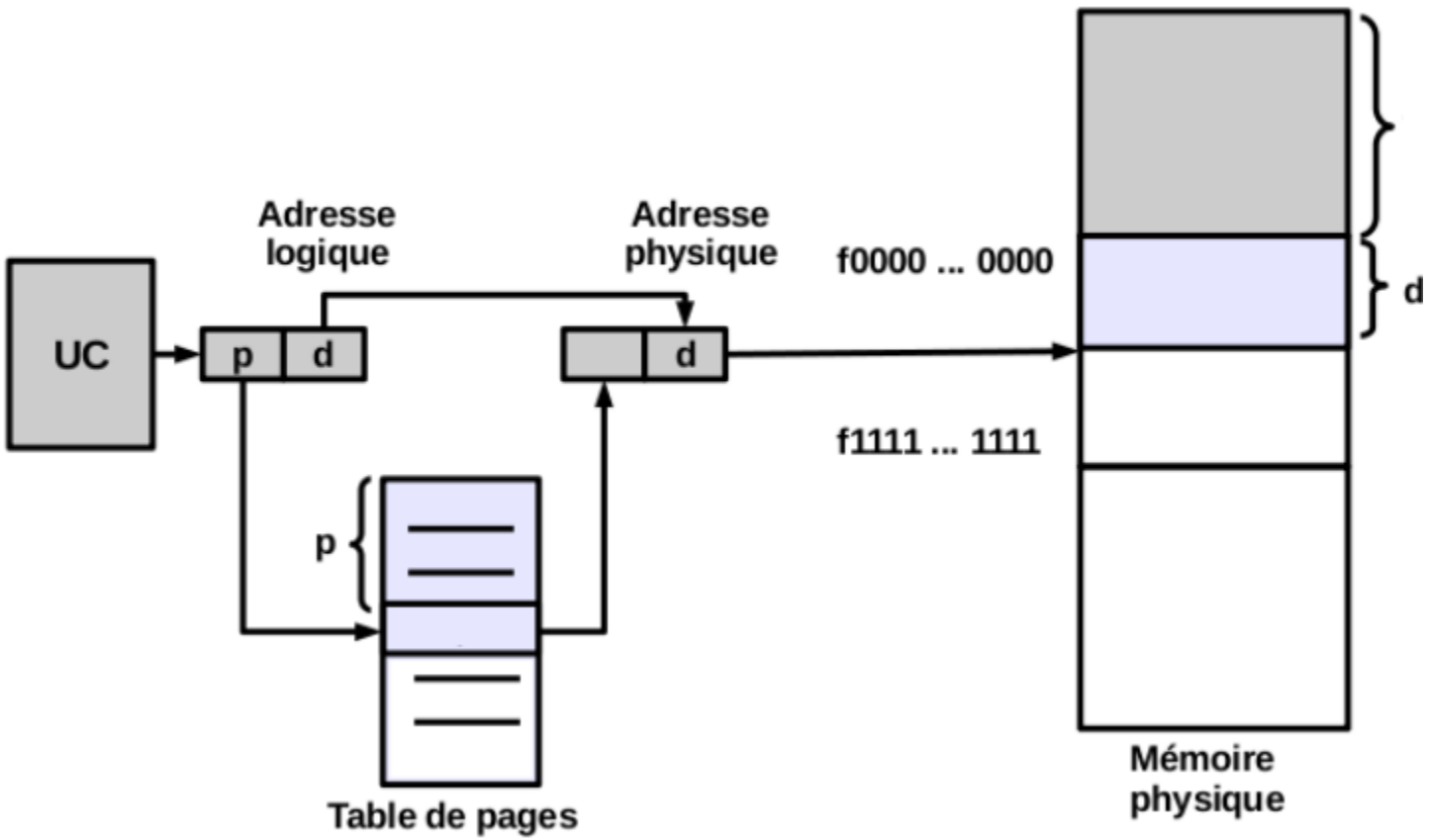
Traduction des adresses virtuelles en adresses physiques

- Il est à rappeler que la **traduction** d'une adresse virtuelle en adresse physique est réalisée par un dispositif matériel appelé "**unité de gestion mémoire**" (MMU).
- Dans un système à mémoire paginée **l'adresse virtuelle** est composée de :
  - Un numéro de page virtuelle (**p** bits de poids forts: gauche),
  - Un déplacement à l'intérieur de la page (**d** bits de poids faibles: droite).

- Taille de la page = 2^{d}
- Chaque adresse générée par l'UC (sur m bits) est structurée comme suit :



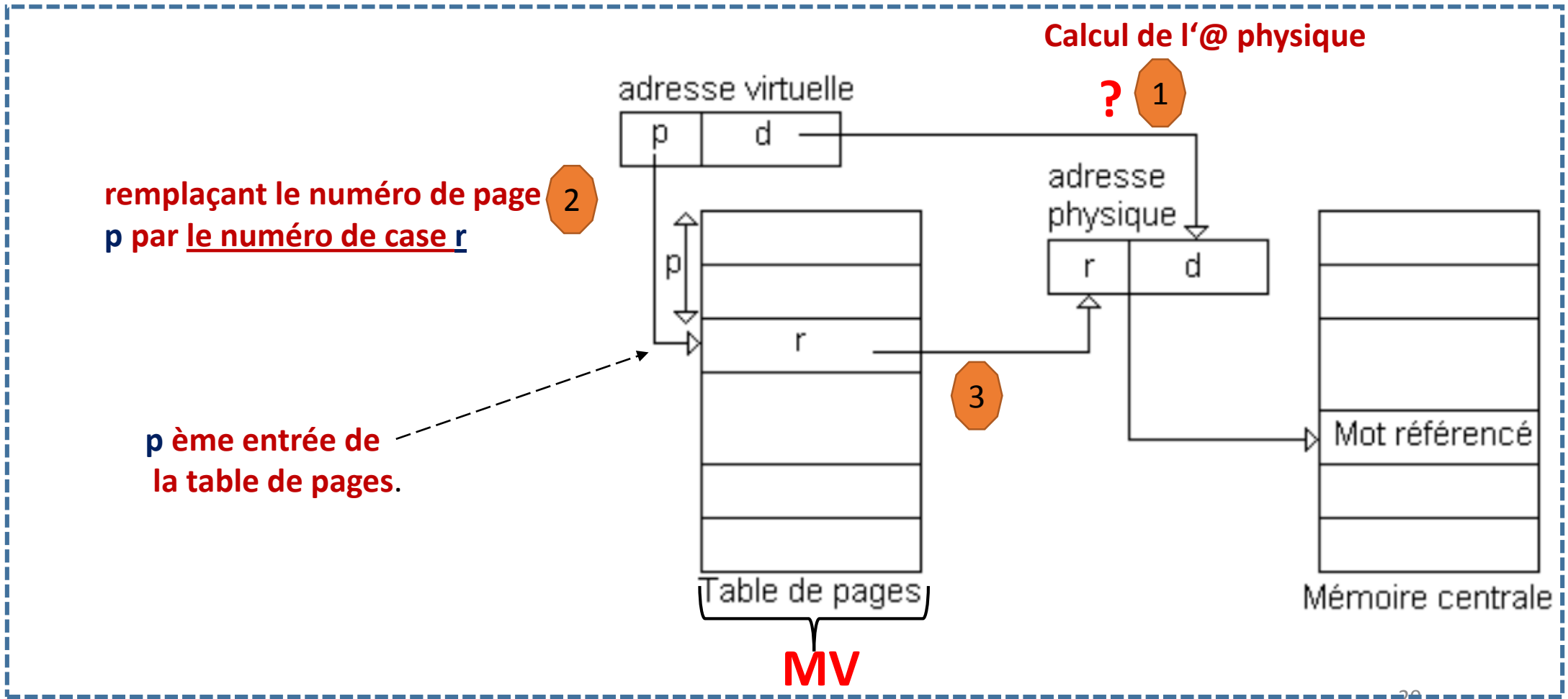
Mémoire virtuelle (Cont.)



Principe de la pagination

Établir la correspondance entre <page virtuelle> et <page physique> (ou case)

- On utilise une table que l'on appelle **table de pages**.
  - **Contient le numéro de case (adresse physique)**



Exercice: AL -> AP

Soit une mémoire physique de **32 octets** ; la taille d'une page est de **4 octets.** On suppose une architecture à **8 bits** (un mot=un octet). Un processus se présente comme suit:

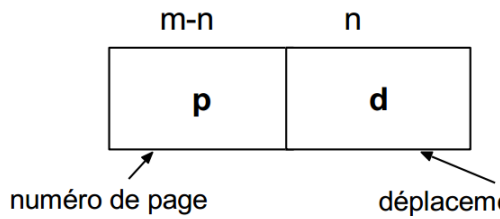
Convertir les adresses logiques suivantes en adresses physiques.

AL=0 ► p=0, d=0 soit AP = 5,0 = 5x4 + 0 = 20

AL=3 ► p=0, d=3 soit AP = 5,3 = 5x4 + 3 = 23

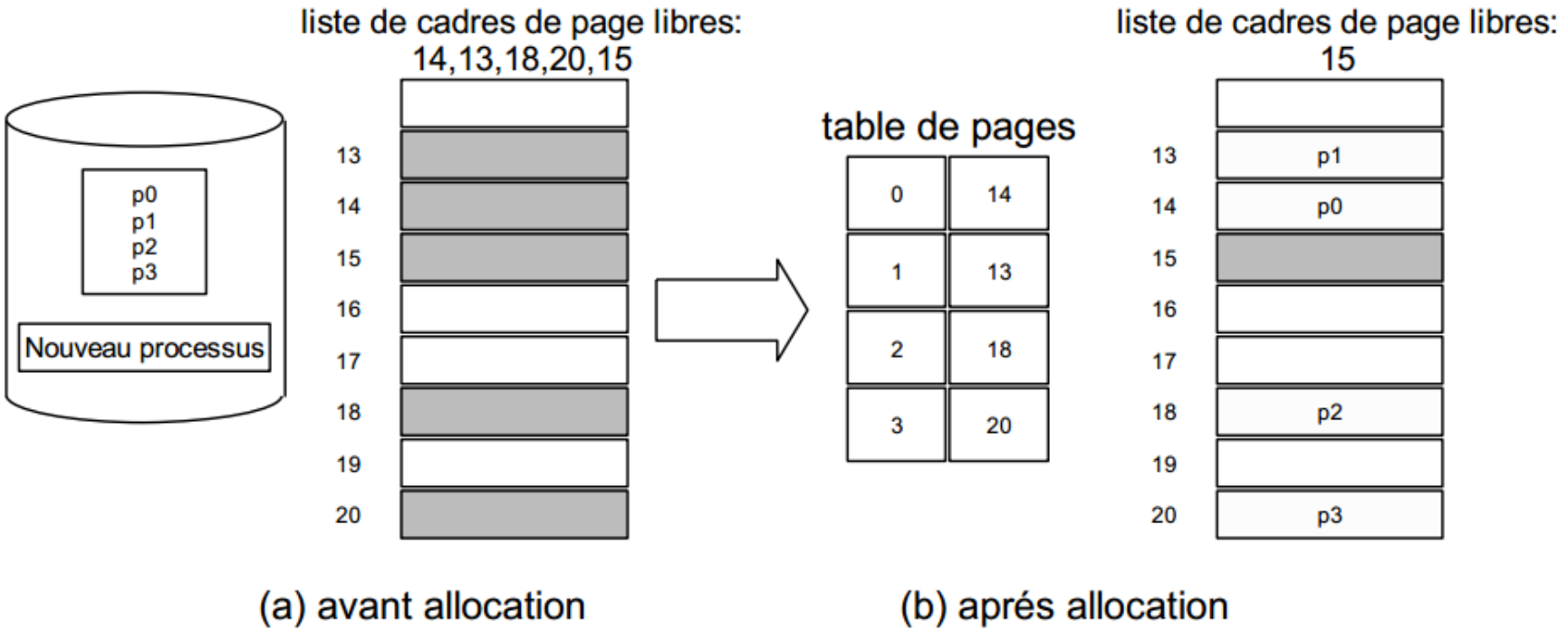
AL=4 ► p=1, d=0 soit AP = 6,0 = 6x4 + 0 = 24

AL=13 ► … AP = 9



- Taille de la page = 2^{d}

Processus de pagination



(a) avant allocation (b) aprés allocation

- le système dispose d'une table de cadres de page qui indique l'état (libre ou occupé) du cadre de page, occupé par quelle page, de quel processus.

Protection de la mémoire paginée

A chaque page sont associés des bits de protection qui font partie de table de pages. La protection peut être :

- Page en lecture/écriture
- Page en lecture seule
- Page en lecture/exécution.

Exemple: IBM/370 : 4 bits/page

Partage du code et des données (partage de pages)

- La pagination permet de partager, entre plusieurs processus, des pages de code ou de données.

Gestion de la mémoire virtuelle : La pagination à la demande

- **La stratégie de chargement :** Quand le système charge-t-il les pages en mémoire principale ?

- **La stratégie de remplacement :** Comment le système choisit-il les pages qui doivent être retirées de la mémoire principale, lorsque aucune case n'est plus disponible ? Pour le choix d'une page, on peut tenir compte des informations sur l'utilisation passée des pages présentes en mémoire principale.

Représentation des espaces virtuels des processus et de l'espace physique

Gestion de la mémoire virtuelle : La pagination à la demande

- **La stratégie de chargement :** Quand le système charge-t-il les pages en mémoire principale ?

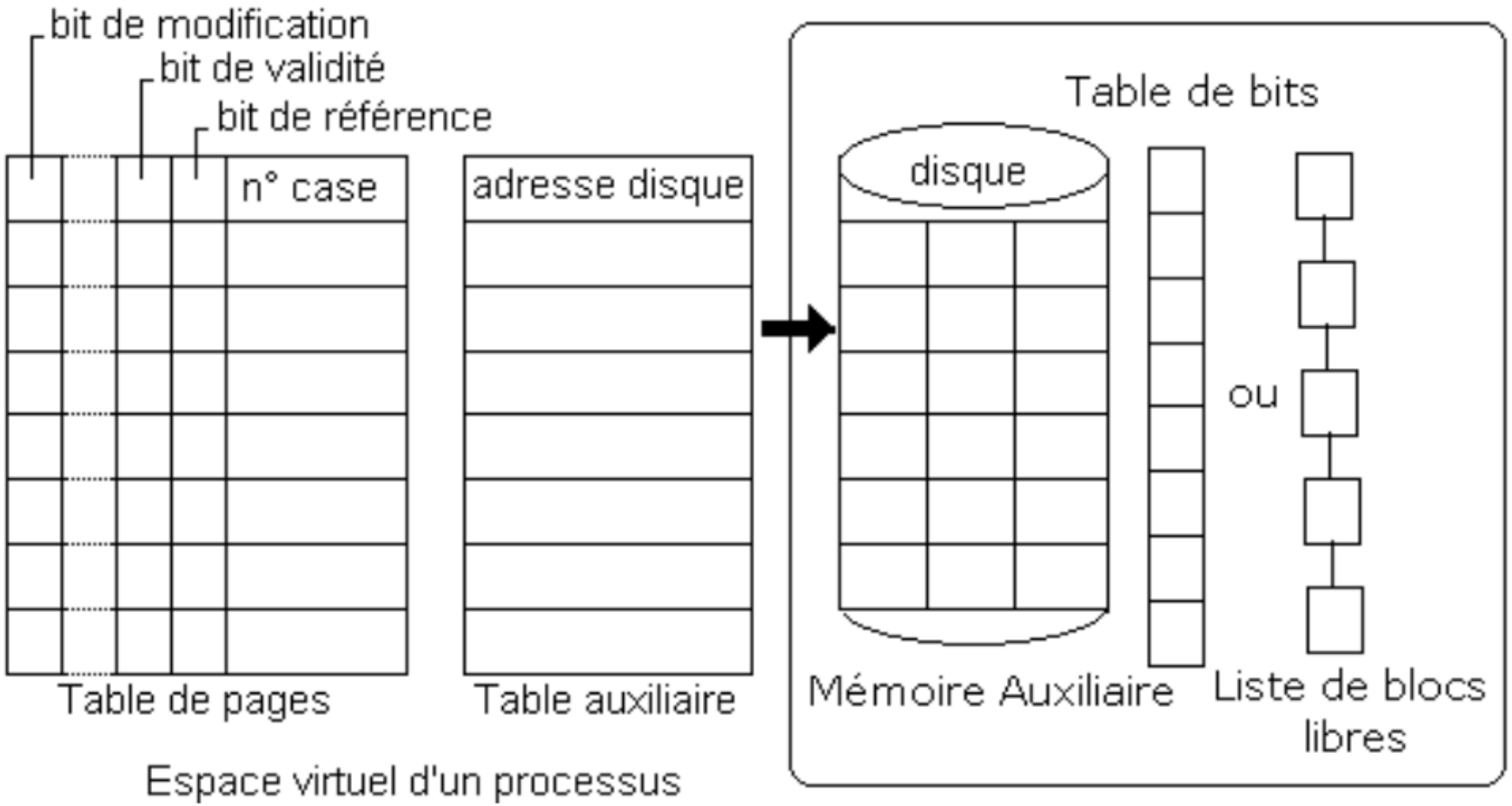
- **La stratégie de remplacement :** Comment le système choisit-il les pages qui doivent être retirées de la mémoire principale, lorsque aucune case n'est plus disponible ? Pour le choix d'une page, on peut tenir compte des informations sur l'utilisation passée des pages présentes en mémoire principale.

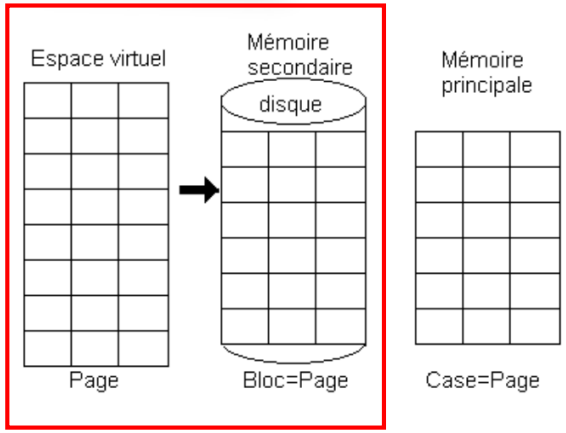
Représentation des espaces virtuels des processus

L'espace virtuel d'un processus est défini par:

- la table de pages,
- la table auxiliaire ou secondaire
- et la Mémoire secondaire ou zone de swap.

Représentation des espaces virtuels des processus

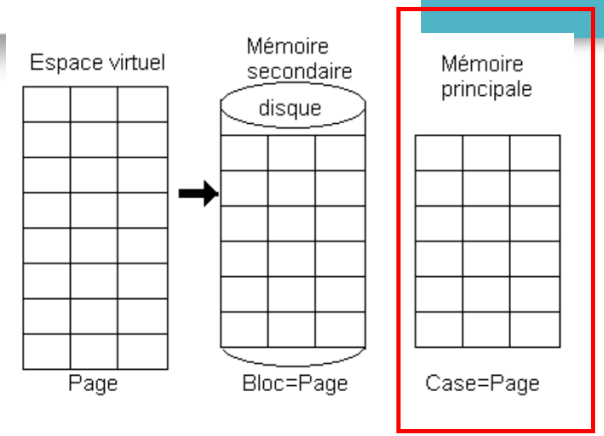


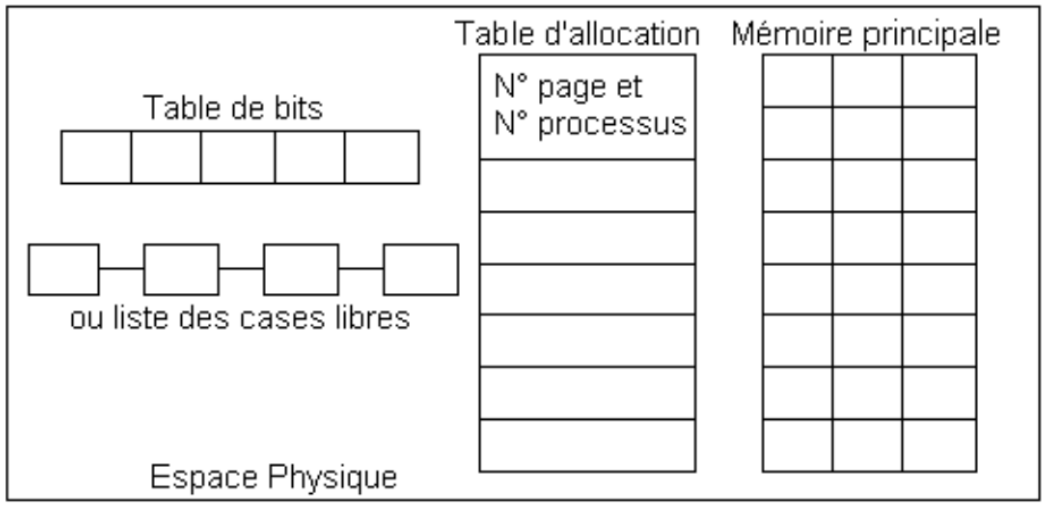


Représentation de l'espace physique

L'espace physique est représenté à l'aide d'une table que **l'on appelle table d'allocation.**

Cette table a autant d'entrées que de cases dans la mémoire principale (en plus de cette table, on utilise une table de bits : un bit par case ou une liste des cases libres). Lorsqu'une case est allouée à une page virtuelle d'un processus donné; on met, dans l'entrée correspondante de la table d'allocation, le numéro de cette page et le numéro du processus auquel elle appartient ou l'adresse de l'entrée de la table auxiliaire qui correspond à cette page





Représentation des espaces virtuels des processus

L'espace virtuel d'un processus est défini par:

- la table de pages,
- la table auxiliaire ou secondaire
- et la Mémoire secondaire ou zone de swap.

a) La table de pages:

Elle est utilisée directement par le mécanisme de traduction des adresses (MMU).

Une entrée de la table de pages contient des informations telles que :

- Numéro de case mémoire correspondant à la page;
- Bits de protection : page en lecture/écriture, en lecture, lecture/exécution; les les bits de protection peuvent être utilisés comme « **Bit de validité** » : page utilisée ou pas par le processus (un processus utilise une partie seulement de l'espace virtuel);
- Bit de présence : page chargée en mémoire centrale ou non;
- Bit de Modification : page modifiée ou non depuis le dernier chargement;
- Bit de référence : page référencée ou non.

Représentation des espaces virtuels des processus

- Lorsqu'un **défaut de pages** se produit, le système doit charger la page correspondante depuis la mémoire secondaire (disque) dans une case de la mémoire principale.
- Il doit donc disposer d'une table que l'on appelle **table auxiliaire** qui décrit l'image disque de l'espace virtuel d'un processus.
- On peut ajouter une adresse disque par entrée (page) de la table de pages.

Représentation des espaces virtuels des processus

L'espace virtuel d'un processus est défini par:

- la table de pages,
- la table auxiliaire ou secondaire
- et la Mémoire secondaire ou zone de swap.

b) La table auxiliaire ou secondaire

Elle contient autant d'entrées qu'il y a de pages dans l'espace virtuel considéré. Chaque entrée de cette table donne l'adresse de la page en mémoire secondaire réservée à la pagination (adresse disque). Cette table est également utilisée lorsque l'on doit recopier une case contenant une page modifiée.

A un instant donné, il y a autant de couples (table de pages, table auxiliaire) que d'espaces virtuels définis. Un seul de ces espaces est accessible au processeur (U.C.) : l'espace du processus actif.

Représentation des espaces virtuels des processus

L'espace virtuel d'un processus est défini par:

- la table de pages,
- la table auxiliaire ou secondaire
- et la Mémoire secondaire ou zone de swap.

Représentation des espaces virtuels des processus

c) Mémoire secondaire ou zone de swap (ou disque de pagination) :

Elle est découpée en blocs de même taille que les pages et contenant l'image de l'espace virtuel (programmes, les données, piles) des processus.

Détection et traitement d'un défaut de page

- **Détection de défaut de page**
- Un défaut de page est détecté lors de la traduction d'une adresse virtuelle en adresse réelle, en accédant à la table de pages et en testant le bit de présence (test réalisé par le MMU) :
- Bit de présence = 1 : prendre le numéro de case contenu dans cette entrée.
- Bit de présence = 0 : la page n'est pas présente en mémoire centrale, donc déroutement pour défaut de page.
- **Traitement des défauts de page**

Remplacement de pages

Quand il y a un défaut de page, on doit allouer une case pour charger la page référencée. S'il n'existe pas de page, on choisit une parmi celles qui sont présentes en mémoire et on affecte sa case à la page référencée. La page sélectionnée au moyen d'un algorithme de remplacement est appelée "**page victime**".

- **Les algorithmes de remplacement**

Algorithme FIFO

C'est l'algorithme de remplacement le plus simple. Quand on veut remplacer une page, on choisit la page la plus ancienne

Implémentation

Chaque fois que l'on charge une page en mémoire centrale, on met (on enfile) le numéro de cette page dans une file gérée en FIFO (premier entré premier sorti).

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	2		2	2	4	4	4	0			0	0			7	7	7
	0	0	0		3	3	3	2	2	2			1	1			1	0	0
		1	1		1	0	0	0	3	3			3	2			2	2	1
X	X	x	x		x	x	x	x	x	x			x	x			x	x	x

15 défauts de pages (x = défaut de page)

- **Les algorithmes de remplacement**

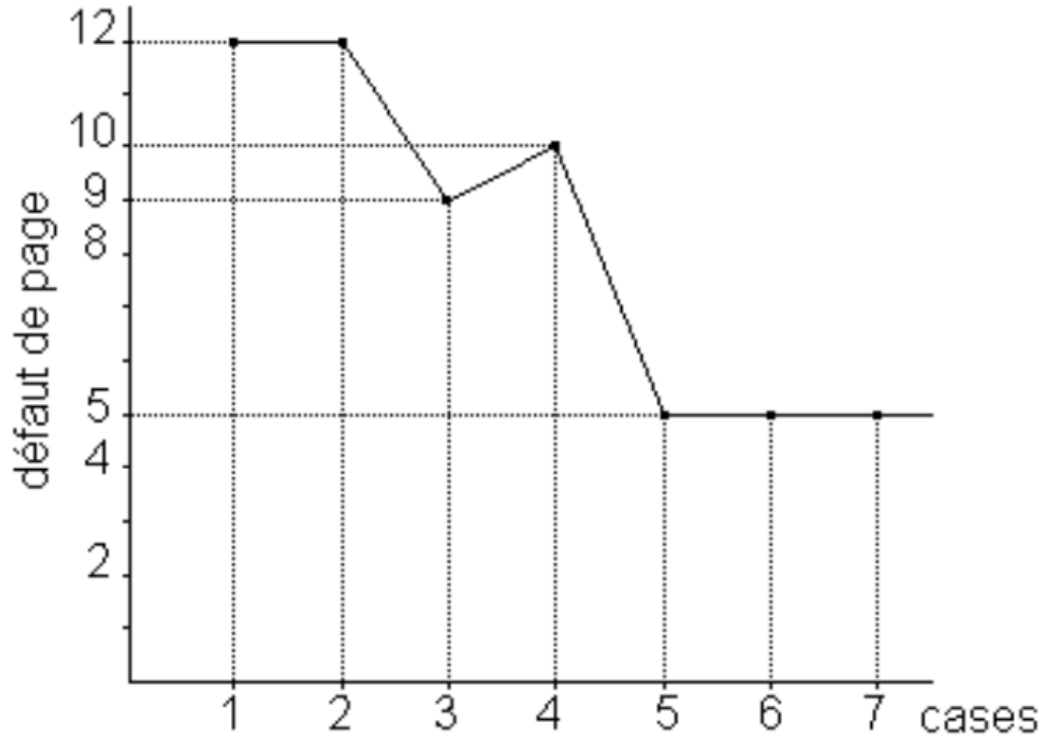
Algorithme FIFO

L'anomalie de belady

Considérons la chaîne de référence suivante : Ω : 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5.

On peut constater sur la figure ci-après que le nombre de défauts de page f(Ω , 4) est supérieur à celui de f(Ω ,3): f(Ω ,4)=10 et f(Ω ,3)=9. C'est l'anomalie de Belady : le nombre de défauts de page croit quand le nombre de cases augmente

(f(Ω ,4) > f(Ω ,3)).



• Exercice-FIFO

•If FIFO page replacement is used with four page frames and eight pages, how many page faults will occur with the reference string

0 1 7 2 3 2 7 1 0 3

•if the **four frames** are initially empty?

Answer: FIFO yields 6 page faults

- **Les algorithmes de remplacement**

L'algorithme optimal (OPT)

Un algorithme de remplacement optimal minimise le taux de défauts de page. Cet algorithme choisit la page qui ne sera plus utilisée ou la page qui sera utilisée le plus tard possible. Cet algorithme suppose une connaissance de l'ensemble de la chaîne de références Ω; il est donc irréalisable en temps réel. Il permet d'évaluer, par comparaison, les autres algorithmes.

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1	7
7	7	7	2		2		2			2			2				7			
	0	0	0		0		4			0			0				0			
		1	1		3		3			3			1				1			
x	x	x	x		x		x			x			x				x			

9 défauts de pages

- # Les algorithmes de remplacement

L'algorithme optimal (OPT)

Considérons la chaîne de référence suivante :

Ω : 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5 avec m=3 cases on aura 7 défauts de pages

1	2	3	4	1	2	5	1	2	3	4	5
1	1	1	1			1			3	3	
	2	2	2			2			2	4	
		3	4			5			5	5	
x	x	x	x			x			X	x	

Ω : 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5 avec m=4 cases on aura 6 défauts de pages.

1	2	3	4	1	2	5	1	2	3	4	5
1	1	1	1			1				4	
	2	2	2			2				2	
		3	3			3				3	
			4			5				5	
x	x	x	x			x				x	

L'algorithme Opt (ou min) **ne comporte l'anomalie de Belady** car le nombre de défauts de pages de f(Ω ,3) est supérieur à celui de f(Ω,4); f(Ω ,3)=7 et f(Ω ,4)=6;

- **Les algorithmes de remplacement**

Algorithme LRU (Least Recently Used

L'algorithme LRU (moins récemment utilisée) choisit la page qui n'a pas été utilisée depuis longtemps (ou la page ayant fait l'objet d'une référence la plus tardive). Cette classe d'algorithme est appelée algorithme à pile ou «stack algorithms ».

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	2		2		4	4	4	0			1		1		1		
	0	0	0		0		0	0	3	3			3		0		0		
		1	1		3		3	2	2	2			2		2		7		
x	x	x	x		x		x	x	x	x			x		x		x		

x : défaut de page

12 défauts de pages

• Les algorithmes de remplacement

Algorithme LRU (Least Recently Used

LRU associe à chaque page la dernière date d'utilisation (date logique). Considérons la chaîne de référence suivante :

ω : 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5 avec m=3 cases ➔ 10 défauts de pages

1	2	3	4	1	2	5	1	2	3	4	5
1	1	1	4	4	4	5			3	3	3
	2	2	2	1	1	1	1		1	4	4
		3	3	3	2	2		2	2	2	5
X	X	x	x	x	x	x			x	x	x

ω : 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5 avec m=4 cases ➔ 8 défauts de pages.

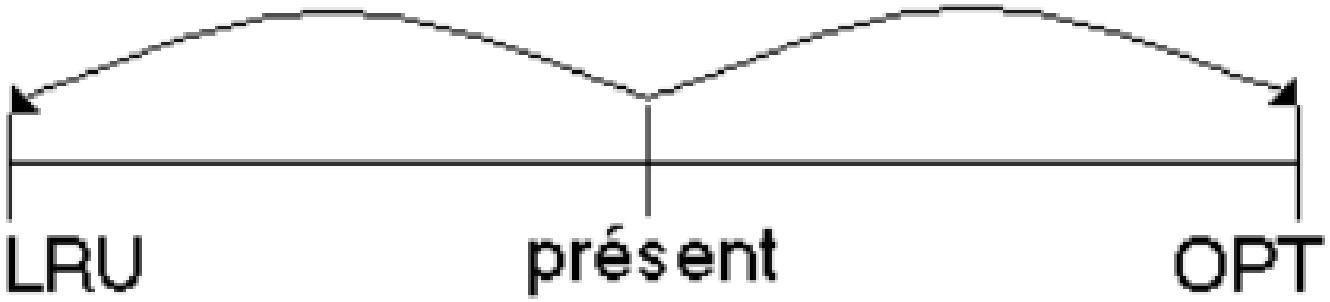
1	2	3	4	1	2	5	1	2	3	4	5
1	1	1	1	1		1	1		1	1	5
	2	2	2		2	2		2	2	2	2
		3	3			5			5	4	4
			4			4			3	3	3
X	X	x	x			x			x	x	x

L'algorithme LRU ne comporte pas l'anomalie de Belady car le nombre de défauts de pages de **f(ω ,3)** est supérieur à celui de **f(ω,4)**; **f(ω,3)=10 et f(ω,4)=8**;

LRU associe à chaque page la dernière date d'utilisation (date logique).

• Comparaison de LRU et OPT

LRU choisit la page qui n'a pas été référencée depuis longtemps. OPT choisit la page qui ne sera pas référencée pendant longtemps (qui sera référencée le plus tard possible).



• Exercice -LRU

- If LRU page replacement is used with four page frames and eight pages, how many page faults will occur with the reference string

0 1 7 2 3 2 7 1 0 3

- if the **four frames** are initially empty?

Answer: LRU yields 7 page faults

• Exercice: Implémentation de LRU

En général, on utilise une pile. Une approche consiste à garder les numéros de pages référencées dans une pile.

Implémentation de la pile

La pile est implémentée à l'aide d'une liste avec double chaînage. On utilise deux pointeurs:

- Un pointeur « sommet » pour empiler.
- Un pointeur « base » pour retirer un élément de la base

• Les algorithmes de remplacement

autre algorithmes

- L'algorithme de l'horloge
- LFU (ou NFU): Least Frequently Used/moins fréquemment utilisée
- L'algorithme du vieillissement (Aging)
- Les algorithmes de remplacement Algorithme de seconde chance