

Cours de systèmes d'exploitation centralisés

Chapitre 2

Gestion de la mémoire virtuelle

Séance 1

N. TEMGLIT , M.RAHMANI

© 2023-2024

Gestion de la mémoire virtuelle

1. Introduction

1.1 Définition

1.2 Adresses logiques (virtuelles) et adresses physiques

1.3 Espace d'adressage logique(virtuel) et espace d'adressage physique

1.4 Principaux objectifs du concept de mémoire virtuelle

2. Pagination

2.1 Implémentation de la mémoire virtuelle

2.2 Traduction des adresses virtuelles en adresses physiques

2.2.1 Adresse virtuelle

2.2.2 Table des pages

2.2.3 Calcul de l'adresse physique

2.2.4 Calcul de la taille de la table de pages d'un processus

2.2.5 Implantation de la table de pages

2.3 Pagination à plusieurs niveaux

2.4 Table de pages inverse

2.5 Choix de la taille des pages

2.6 Mémoire associative ou registres associatifs (TLB)

2.7 Protection de la mémoire paginée

2.8 Partage du code et des données (partage de pages)

3. La segmentation

3.1 Représentation de l'espace virtuel d'un processus

3.2 Allocation de la mémoire centrale aux segments

3.3 Traduction d'une adresse virtuelle en adresse réelle

3.4 Implémentation de la table de segments

3.5 Protection des segments

3.6 Partage de segment

4. Segmentation avec pagination

4.1 Traduction d'une adresse virtuelle en adresse physique

4.2 Exemples

4.2.1 Segmentation et pagination de l'Intel 80x86(32bits sans PAE)

4.3.2 Pagination dans le système Linux jusqu'à la version 2.6.10

4.3.2 Pagination 4 niveaux de Linux depuis la version 2.6.11

1. Introduction

- Gestion de la mémoire principale(rappel) :

Système en multiprogrammation

- **Partitions fixes** : Fragmentation interne et externe.
- **Partitions variables** : fragmentation externe.

1.1 Définition

- La mémoire virtuelle :
 - C'est l'ensemble des emplacements désignés par les adresses générées par le processeur.

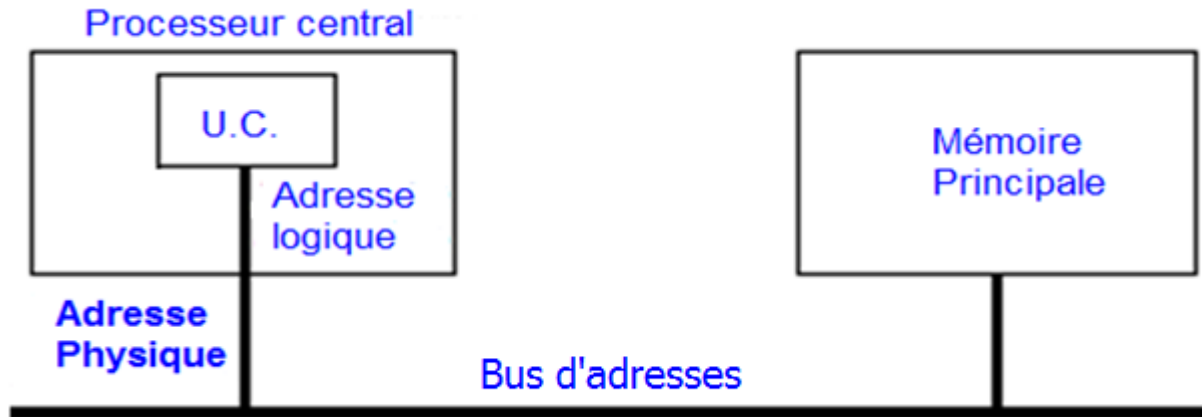
1.2 Adresses logiques (virtuelles) et adresses physiques

- Les adresses générées par le processeur sont appelées : **adresses logiques ou adresses virtuelles**.
- Les adresses générées par l'unité de gestion mémoire (MMU: Memory Management Unit) sont appelées : **adresses physiques ou adresses réelles**: Ces adresses correspondent à des emplacements de la mémoire principale(RAM).

1.3 Espace d'adressage logique (virtuel) et espace d'adressage physique

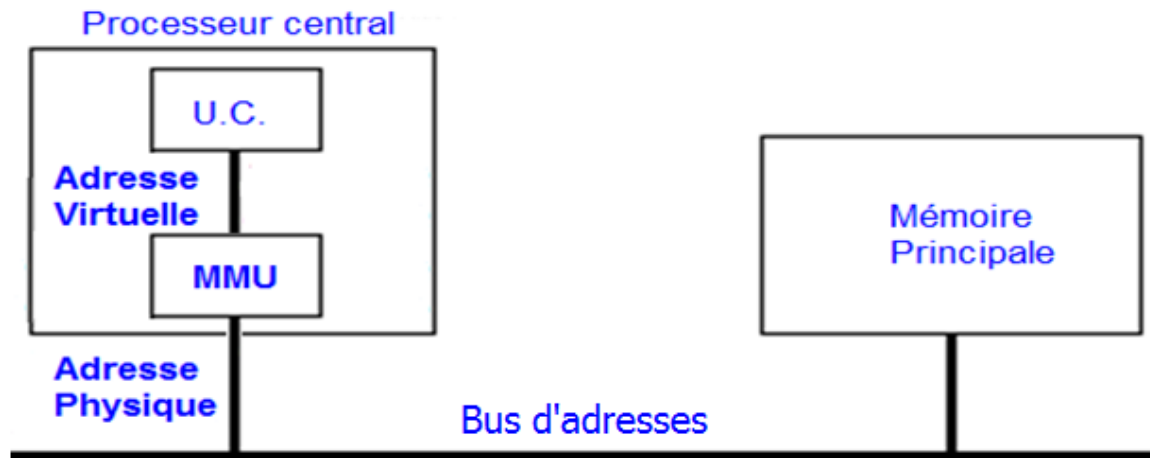
- **Espace d'adressage logique (virtuel) d'un processus :**
C'est l'ensemble de toutes les adresses logiques (virtuelles) générées au cours de l'exécution de ce processus.
- Cet ensemble est limité par la taille du registre adresse.
- **Espace d'adressage physique:**
C'est l'ensemble de toutes les adresses physiques (réelles) correspondant aux adresses virtuelles de l'espace d'adressage virtuel du processus.

Exemple1: Machine sans mémoire virtuelle



- Sur une machine sans mémoire virtuelle, l'adresse générée par le processeur est directement placée sur le bus d'adresse, sans aucune modification → **adresse logique = adresse réelle.**

Exemple2: Machine avec mémoire virtuelle



- Lorsque la mémoire virtuelle est utilisée → les adresses virtuelles ne sont pas directement placées sur le bus :
 - Elles sont envoyées à l'unité de gestion mémoire (MMU).
 - Le MMU traduit les adresses virtuelles en adresses réelles (ou physiques) et place ces adresses physiques sur bus.
- La traduction des Adresses virtuelles en adresses réelles est réalisée, par le MMU, pendant l'exécution du programme.

1.4 Principaux objectifs du concept de mémoire virtuelle

Le concept de mémoire virtuelle permet :

1. Au programmeur d'écrire des programmes sans tenir compte des contraintes de la taille de la mémoire centrale (RAM).
2. A un processus dont la taille du programme est supérieure à la taille de la mémoire principale de s'exécuter.

Pour satisfaire ces 2 deux objectifs :

- a. Séparation (dissociation) de l'adressage virtuel (logique) et de l'adressage physique (réel):
 - L'espace d'adressage virtuel dépend de la taille du registre adresse.
- b. Découpage (fractionnement) de l'espace d'adressage virtuel d'un processus en blocs (segments, pages, ...).
- c. Allocation partielle de l'espace physique à un processus.
 - ➔ Un processus peut commencer son exécution sans que son programme soit entièrement chargé en mémoire centrale.

Les points (b) et (c) ➔ Un processus peut s'exécuter même si la taille de son programme est supérieure à celle de la mémoire centrale.

- L'espace d'adressage réel ou physique dépend de la taille de la mémoire principale (RAM) existante sur la machine
- La taille maximale de la mémoire réelle dépend de la taille du bus d'adresses.
- **L'espace d'adressage virtuel est indépendant de l'espace d'adressage réel.**
- Un processus peut donc adresser un espace virtuel beaucoup plus grand que la mémoire centrale.

Exemple1:

- Considérons une machine ayant une adresse virtuelle sur **16 bits** et disposant de **4096 mots** mémoire.
- Un programme peut adresser 65536 (2^{16}) mots.

- L'espace d'adressage logique de cette machine est formé des adresses 0, 1, 2, ..., 65635.
- Alors que son espace d'adressage physique comprend les adresses 0, 1, 2, ..., 4095.
- **Les deux espaces sont de tailles différentes.**

Implémentation de la MV

2. Pagination

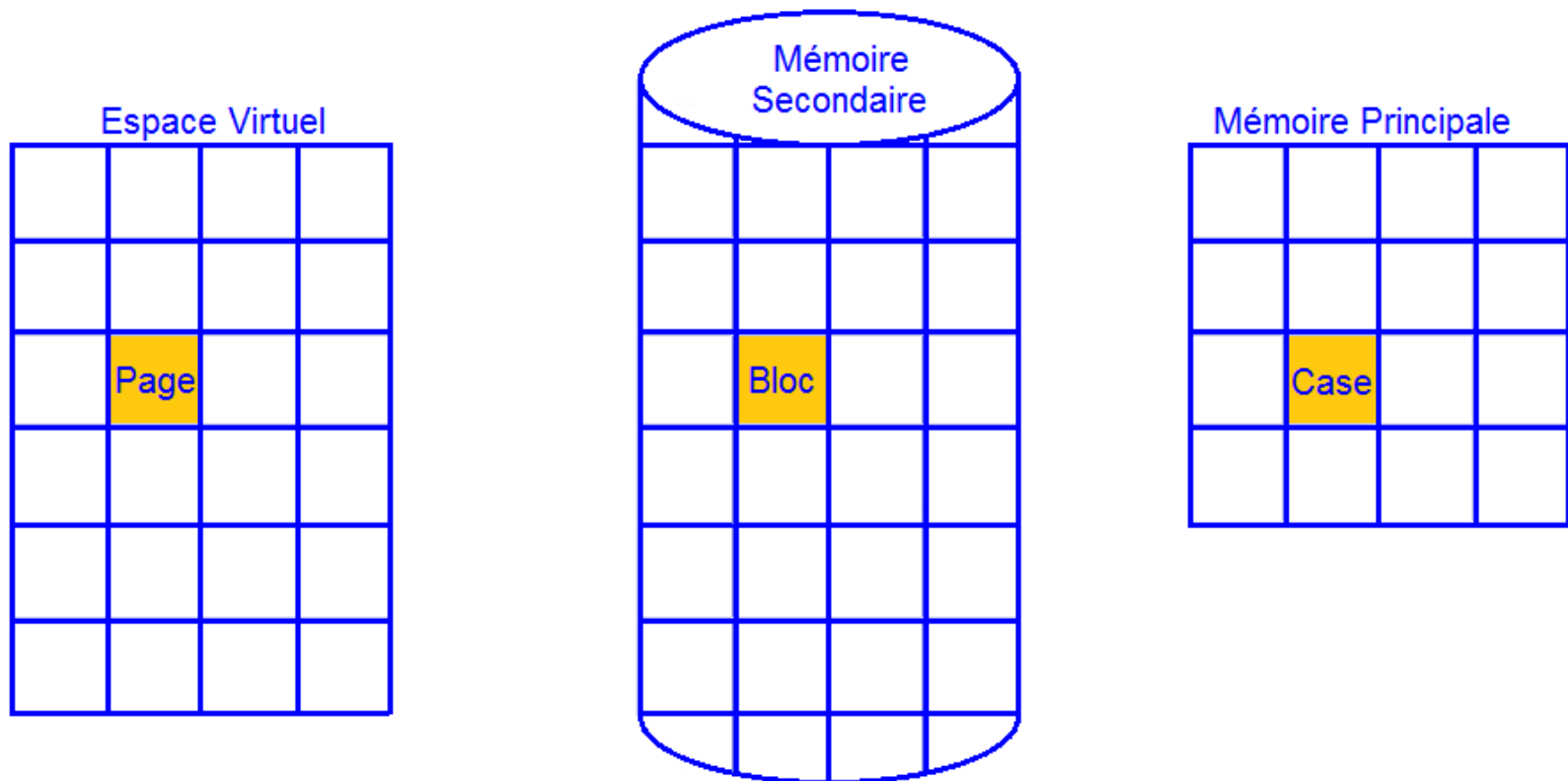
2.1 Implémentation de la mémoire virtuelle

Solutions matérielles : très efficaces.

Exemples : pagination, segmentation, segmentation avec pagination.

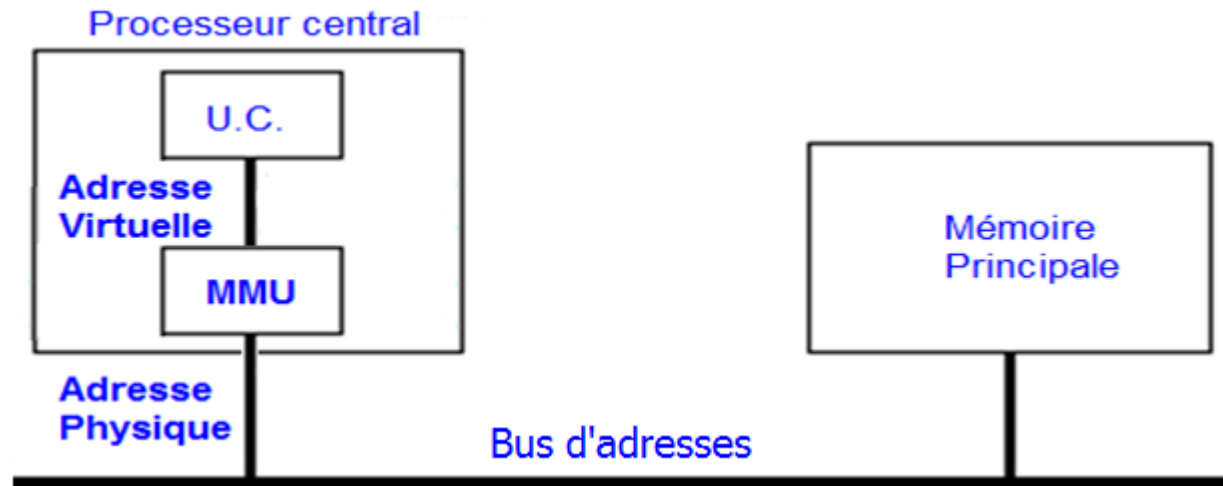
- La pagination consiste à diviser l'espace d'adressage virtuel en unités ou blocs appelés **pages virtuelles** (**page virtuelle = page**).
- La mémoire physique est également découpée en blocs ou unités de même taille que les pages, appelés **cases** , **pages physiques** ou **pages réelles** (pages frames).

- Une mémoire secondaire(disque) est utilisée comme extension de la mémoire principale; on l'appelle disque de pagination.
 - Elle permet de stocker les espaces virtuels (Les pages) des processus.
 - La mémoire secondaire est découpée en **blocs de même taille** que les pages.



2.2 Traduction des adresses virtuelles en adresses physiques

La traduction d'une adresse virtuelle en adresse physique est réalisée par un dispositif matériel appelé "unité de gestion mémoire" (MMU).



2.2.1 Adresse virtuelle

- Dans un système à **mémoire paginée** l'adresse est composée de:
 - Un numéro de page virtuelle (**m bits** de poids forts : gauche),
 - Un déplacement à l'intérieur de la page (**k bits** de poids faibles : droite).



Taille de la page = 2^k octets

2.2.2 Table des pages

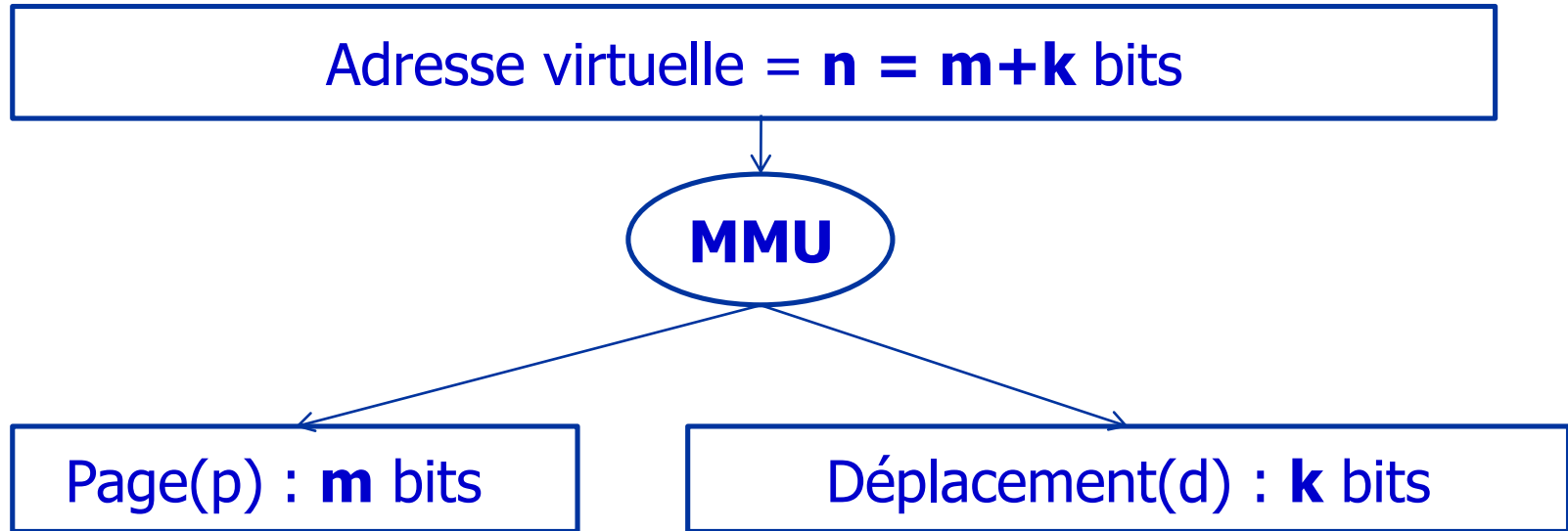
- Pour établir la correspondance entre **page virtuelle** et **page physique** (ou **case**), le MMU utilise une **table de pages**.
- Cette table a un certain nombre d'entrées.
- Chaque entrée contient le numéro de case correspondant à la page **p**.
- Le numéro de page **p** sert d'index dans la table de pages.

Page0	N° Case
Page1	
	2500
	300
Page _{n-1}	

Table de pages

2.2.3 Calcul de l'adresse physique

- Le MMU décompose l'adresse virtuelle (n bits) en :
N° page virtuelle(p) sur m bits et Déplacement(d) sur k bits.

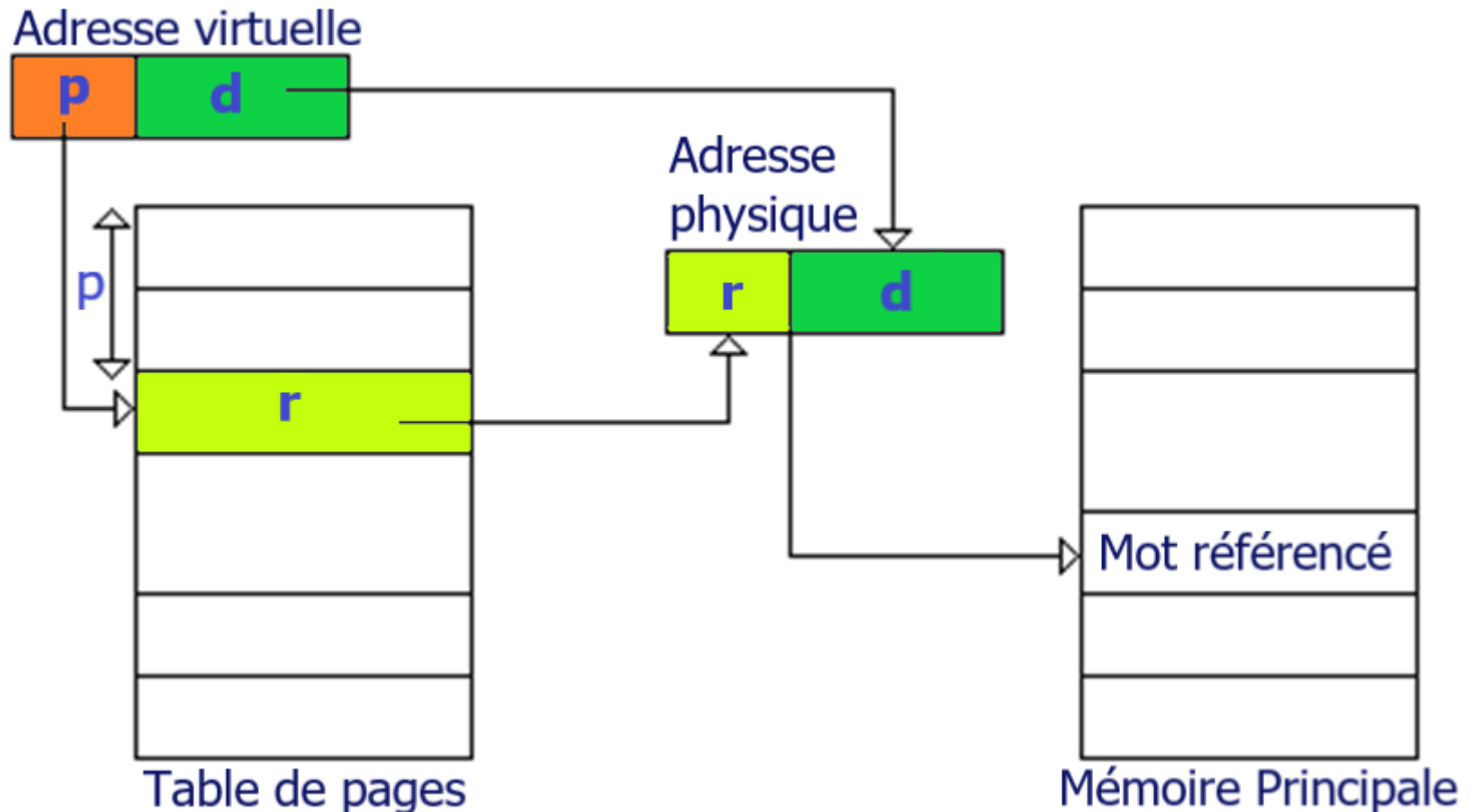


Soient:

- ✓ $[p, d]$ l'adresse virtuelle d'un mot mémoire et
- ✓ $[r, d]$ l'adresse physique correspondant à l'adresse virtuelle $[p, d]$

Comment obtenir l'adresse physique $[r, d]$?

- Le MMU accède à la **p^{ième}** entrée de la table page et
- Affecte à **r** le contenu de l'**entrée p** de la table de page et
- Affecte à **d** la valeur de **d** de l'adresse virtuelle (le déplacement est le même car : taille page = taille case) .



2.2.4 Calcul de la taille de la table de pages d'un processus

- Nombres d'entrées de la table de pages =
Taille de l'espace virtuel / taille d'une page.
- Taille en octets = nombre d'entrées * taille d'une entrée.
- Taille d'une entrée dépend :
 - de la taille maximale de la mémoire physique (bus d'adresse) ,
 - de la taille d'une page et
 - des informations telles que bits de présence, protection , ...

Remarque : La taille d'une entrée doit être un multiple d'octet.

Processeurs Intel 32 bits : entrée = 4 octets.

Processeurs Intel 64 bits : entrée = 8 octets.

- **Exemple :** Taille d'une entrée en ne prenant en compte que la taille maximale de la mémoire principale et la taille d'une page:

Taille d'une entrée (en bits) =

$\log_2(\text{Taille maxi de la mémoire réelle} / \text{taille d'une page})$

MC=4Go et page = 4096 octets →

Taille entrée = $\log_2(2^{32} / 2^{12}) = \log_2(2^{20}) = 20 \text{ bits.}$

- **La table de page décrit l'espace virtuel d'un processus →**
 - Chaque processus a **sa propre** table de pages en mémoire centrale.
 - L'adresse de cette table fait partie du PCB du processus.
 - Les informations transmises au MMU lors de la commutation de contexte dépendent de l'architecture de la machine :
 - a. Table de pages utilisée par le MMU constituée d'un ensemble de registres : La table de pages du processus **élu** est chargée dans la table de pages du MMU.
 - b. Table de pages utilisée par le MMU se trouve en mémoire centrale : l'adresse de la table de pages du processus **élu** et éventuellement sa taille sont chargées dans des registres du MMU (registres du processeur).

La table de page d'un processus peut décrire:

1. Tout l'espace virtuel (max permis par le registre d'adresse):

L'espace utilisé par le processus + l'espace non utilisé par le processus. → **Perte d'espace mémoire.**

2. Uniquement l'espace utilisé par le processus :

Dans la plupart des machines actuelles, la table de pages ne décrit que l'espace réellement utilisé par le processus → **gain d'espace en M.C.**

2.2.5 Implantation de la table de pages

- La table de pages peut être implantée de différentes manières.
- Tenir compte de deux paramètres : **La taille de la table de pages** et **Le temps de la traduction.**

1) Taille de la table de pages

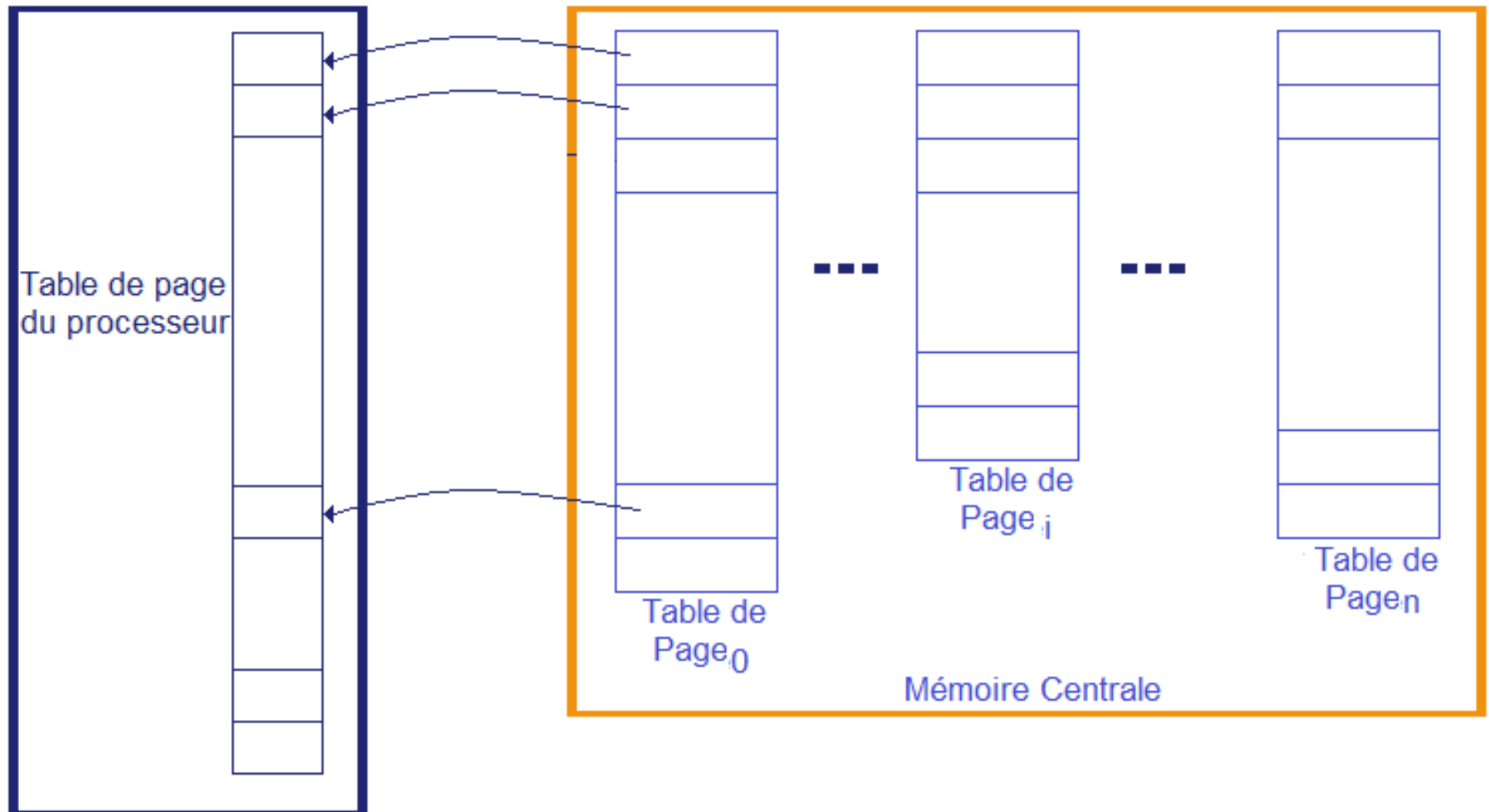
- La taille de table de pages peut être très grande;
- Les ordinateurs actuels utilisent des adresses virtuelles de 32bits ou 64bits.
- **Exemple** : Une machine ayant une adresse de 32bits et des pages de 4k octets (2^{12} octets).
 - ✓ Espace d'adressage virtuel = 2^{32} octets
 - ✓ Nombre d'entrées = $2^{32} / 2^{12} = 2^{20} = 1048576$
 - ✓ Supposons que la taille d'une entrée soit égale à 4 octets.
 - ✓ Taille de la table de pages = $1048576 * 4 = 2^{20} * 2^2 = 4\text{Mo.}$

2) Temps de la traduction

- A chaque référence mémoire → Effectuer une traduction d'adresse donc accéder à la table de pages : La traduction d'adresse virtuelle en adresse réelle doit être rapide.

a. Implantation de la table de pages à l'aide de registres

- La table de pages est constituée d'un ensemble de registres rapides et fait partie du processeur (MMU).
- Les registres ne nécessitent pas d'accès mémoires.
- Chaque processus a sa propre table de pages en mémoire principale
- **La table de pages ne décrit que l'espace réellement utilisé.**
- Lorsqu'un processus est élu, le commutateur de processus (dispatcher) charge la table de pages du processus élu dans la table de pages (ensemble registres) du processeur.



Avantages :

- La table de pages d'un processus ne décrit que l'espace réellement utilisé → **Gain d'espace en mémoire principale.**
- Solution simple à réaliser et ne nécessite aucun accès à la mémoire lors de la traduction des adresses virtuelles en adresses physiques → **Traduction Rapide.**

Inconvénient :

- Le chargement de la table de pages (registres) à chaque commutation peut être pénalisant (chargement réalisé à l'aide d'instructions spéciales):
Si table de pages de grande taille → **Solution coûteuse en temps processeur.**

Calcul de la taille de la table de pages d'un processus

- Nombres d'entrées de la table de pages = Taille de l'espace virtuel réellement utilisé par le processus / taille d'une page.

Exemples: Table de pages constituée de registres

1. Le DEC-PDP/11

- L'adresse est constituée de 16 bits et la taille de la page est de 8192 octets(8k).
- La table de pages est constituée de 8 registres rapides (8 pages de 8 k octets).

2. Le CII-10070

- L'adresse est constituée de 17 bits et
- La taille de la page est de 512 mots de 4 octets.
- La table de pages est constituée de 256 registres.

b. Implantation de la table de pages en mémoire centrale

- La table de pages réside en mémoire principale.
- Cette méthode nécessite deux accès mémoires pour atteindre un mot en mémoire centrale :
 - Un accès à la table de pages,
 - Un accès au mot référencé.

On peut avoir 2 cas :

b-1. La table décrit tout l'espace virtuel

- Le processeur dispose d'**un seul registre** qui pointe sur le début de la table de pages (registre utilisé par le MMU).
- Lors de la commutation de contexte, on ne charge qu'un seul registre avec l'adresse de la table de pages.

Comment distinguer l'espace virtuel (pages) utilisé par un processus de l'espace non utilisé?

- Ajouter un bit que l'on appellera « bit de validité (v) ou des bits de protection » à chaque entrée de la table de pages :
 - Bit $v=1$ ➔ page appartient à l'espace virtuel du processus ;
 - Bit $v=0$ ➔ page n'appartient pas à l'espace virtuel du processus.

1	
1	
1	
1	
0	
	⋮
0	
0	
0	

Espace Virtuel
appartenant au
processus

Espace Virtuel
n'appartenant pas
au processus

Table de Page

- Lors de la traduction d'une adresse le MMU consulte le bit de validité (v) s'il est égal à zéro → Violation de protection mémoire (page n'appartient pas au processus).

Calcul de la taille de la table de pages d'un processus

- Nombres d'entrées de la table de pages = taille maximale de l'espace virtuel / taille d'une page.

b-2. La table de pages ne décrit que l'espace virtuel réellement utilisé

- Dans ce cas le processeur dispose de 2 registres :
 - **Registre adresse** : contient l'adresse de la table de pages,
 - **Registre limite** : contient la taille (nombre d'entrées) de la table de pages.

- Lors d'une commutation de contexte :
 - Registre adresse = adresse de la table de pages du processus élu,
 - Registre limite = taille de la table de page.
- **Exemple** : Le VAX de DEC
- Lors de la traduction d'une adresse le MMU compare le N° de page avec le registre limite:
 - Si $N^{\circ} \text{ page} \geq \text{registre limite} \rightarrow$ violation de protection mémoire (page n'appartient pas au processus).

Calcul de la taille de la table de pages d'un processus

- Nombres d'entrées de la table de pages = taille de l'espace virtuel réellement utilisé par le processus / taille d'une page.

Comment gérer l'espace occupé par les tables de pages en mémoire principale ?

- ✓ La taille de la table de pages ne doit pas être supérieure à une page car l'espace alloué à une table de pages n'est pas contigu (la pagination est appliquée à tous les processus sans distinction).
- ✓ Si la taille de la table de pages est supérieure à celle d'une page, on doit utiliser plusieurs niveaux de pagination (ex: pagination à deux niveaux).

- La taille de la table de pages est proportionnelle à la taille de l'espace virtuel.
- Si l'espace virtuel est très grand → table de pages très grande.
- **Exemple :**

Machine avec adresse virtuelle sur 64 bits et page = 4096 octets.
Nombre d'entrées de la table de pages = $2^{64}/2^{12}=2^{52}$ entrées. Si
taille d'une entrée = 4 octets .

➤ Taille de la table de pages = $2^{52} * 2^2 = 2^{54}$ ou **2^{24} Giga octets.**

2 solutions :

- 1) Pagination à plusieurs niveaux,
- 2) Table de pages inverse.

2.3 Pagination à plusieurs niveaux

- Les tables de pages ne décrivent que l'espace réellement utilisé par un processus(l'espace inutilisé n'est pas décrit).

Exemple : la pagination à deux niveaux.

La pagination à deux niveaux

- L'espace virtuel est divisé en blocs de même taille, que l'on appelle **hyperpages** ou **répertoires** (machine intel et linux).
- Les **hyperpages** (répertoires) sont elles-mêmes divisées en **pages**.
- Une adresse virtuelle est composée de **trois champs** :
 - Numéro d'hyperpage (répertoire) **h**,
 - Numéro de page **p**,
 - Déplacement dans la page **d**.

Adresse Virtuelle

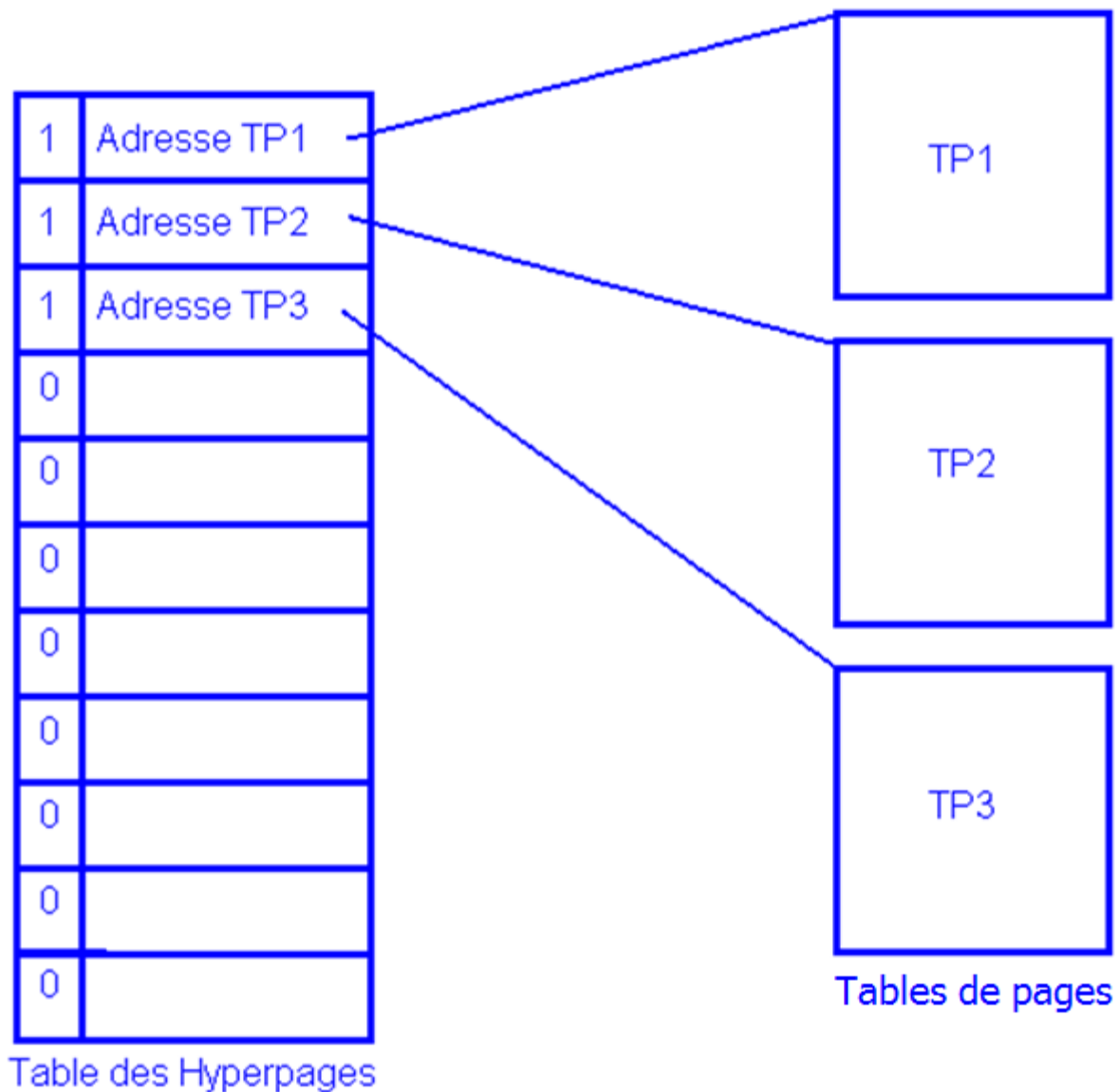
h	p	d
Hyperpage	Page	Déplacement

Hyperpage0 ou Répertoire0				Page 0				Hyperpage4 ou Répertoire 4			
							Page 15				
Page 0				Hyperpage3 ou Répertoire3				Page 0			
			Page 15								Page 15

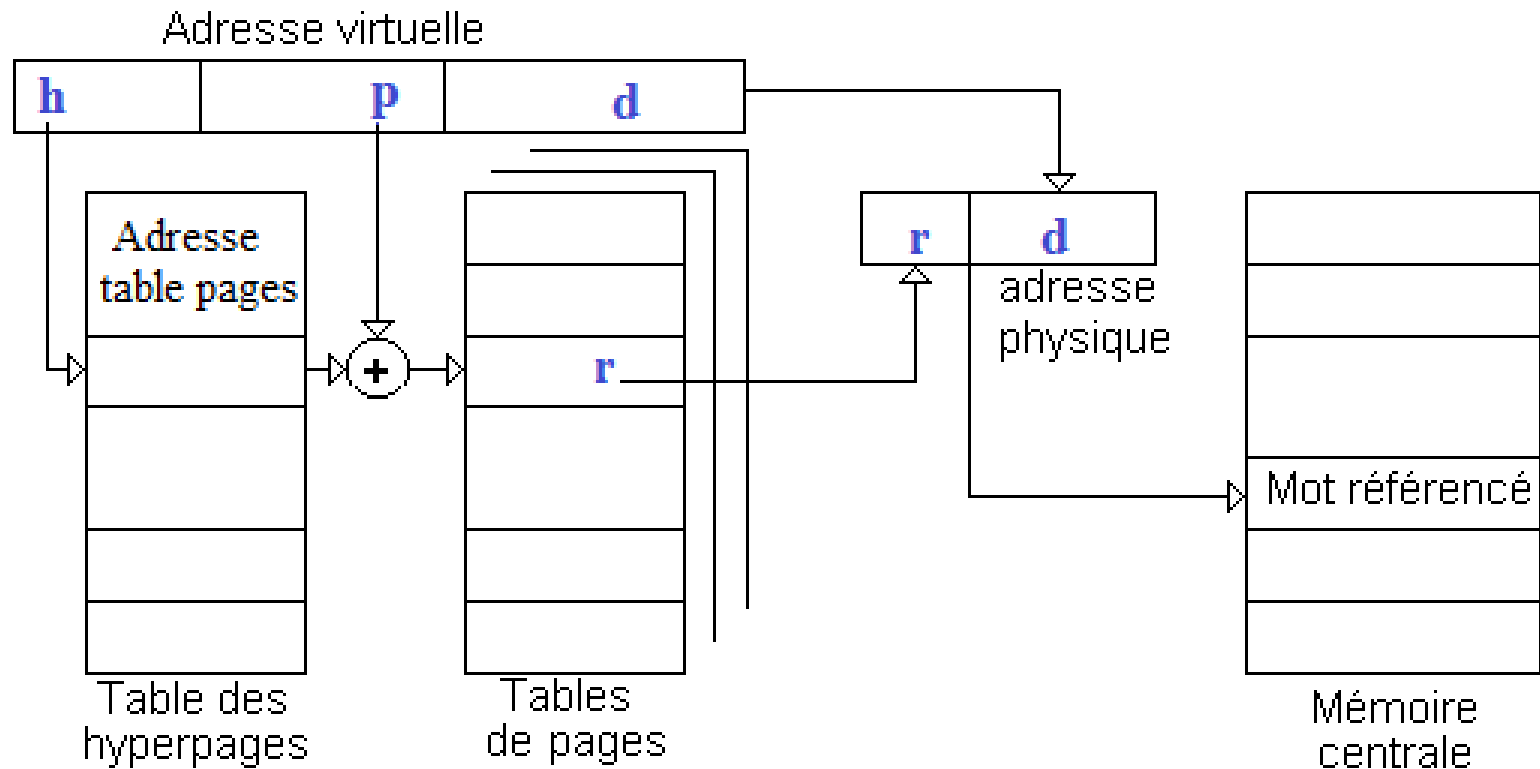
- L'espace virtuel d'un processus est décrit à l'aide :
 - D'une table des hyperpages (répertoires) et
 - D'une ou plusieurs tables de pages.
- Chaque hyperpage (bloc) est décrite à l'aide d'une table de pages
 - ➔ Chaque entrée de la table des hyperpages contient l'adresse ou un pointeur vers sa table de pages.
- Toutes les tables de pages ont la même taille.
- Le numéro de l'hyperpage permet d'accéder à la table des hyperpages.
- La table des hyperpages (répertoires) et les tables de pages d'un processus sont stockées en mémoire centrale.

Remarque :

- Une entrée de la table des hyperpages (répertoires) peut également contenir un "bit de validité" qui indique que l'hyperpage (répertoires) est utilisée ou non par le processus.
- Le MMU peut utiliser un registre contenant la taille de la table des hyperpages (répertoires) , dans ce cas la table des hyperpages ne décrit que l'espace réellement utilisé par le processus (une entrée par table de pages).



Traduction des adresses virtuelles en adresses physiques



- Trois accès sont nécessaires pour atteindre un mot en mémoire centrale:
 1. Accès à la table des hyperpages(répertoires),
 2. Accès à la table de pages,
 3. Accès au mot mémoire.

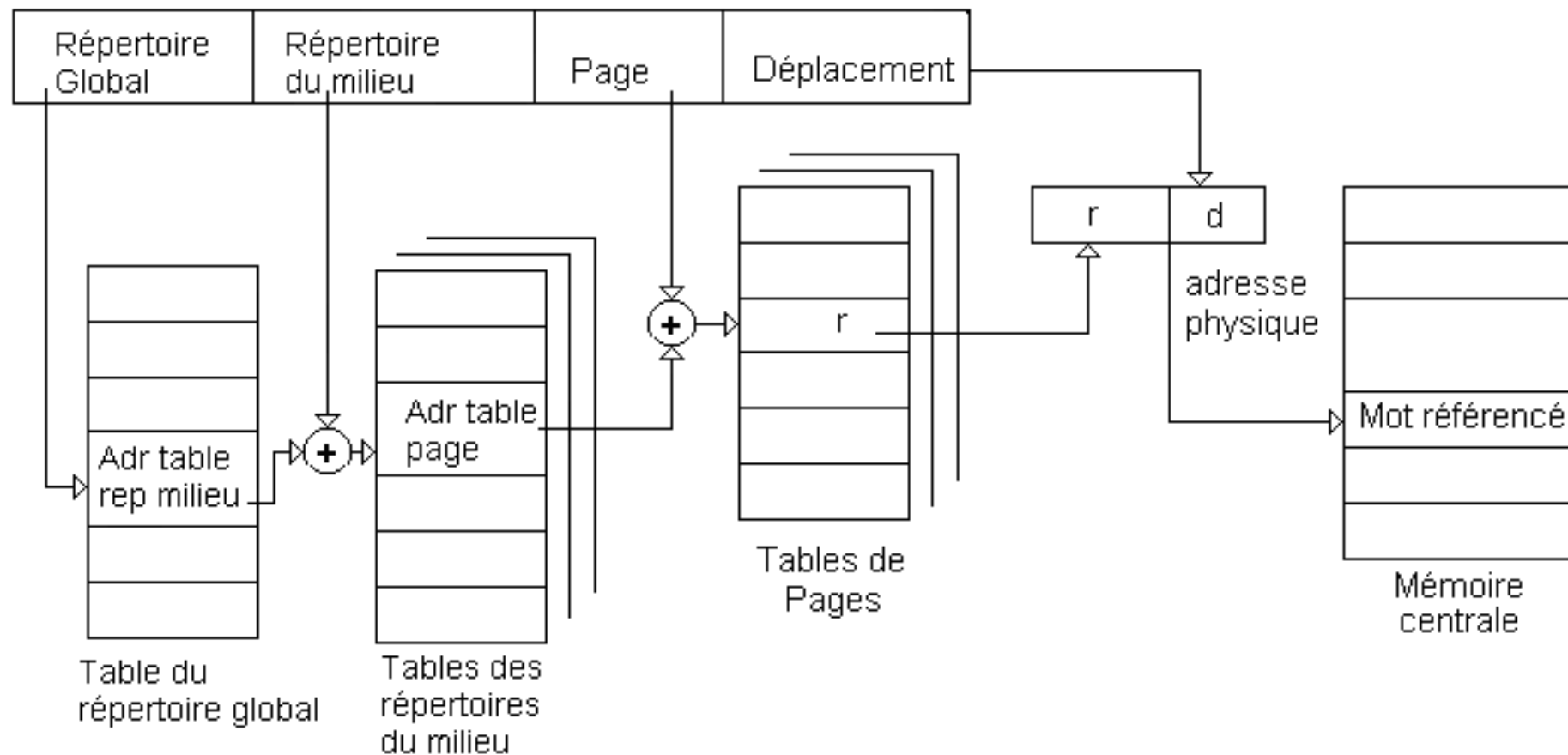
Exemple : Pagination du Système Linux jusqu'à la version 2.6.10

- Le système Linux utilise la pagination à **3 niveaux**
- Une adresse virtuelle est composée de 4 champs :
 - Répertoire global (Page Global Directory),
 - Répertoire du milieu (Page Middle Directory) ,
 - Page (Page Table),
 - Déplacement à l'intérieur de la page.

Adresse virtuelle

Répertoire global	Répertoire du milieu	Page	Déplacement
-------------------	----------------------	------	-------------

Traduction d'une adresse virtuelle en adresse réelle



Exemple3.2 : Pagination du Système Linux depuis la version 2.6.11

- Le système Linux utilise la pagination à **4 niveaux**.
- Une adresse virtuelle est composée de 5 champs:
 - Répertoire global (Global Directory),
 - Répertoire Supérieur(Upper Directory),
 - Répertoire du milieu (Middle Directory) ,
 - Page,
 - Déplacement à l'intérieur de la page.

Adresse virtuelle

Répertoire global	Répertoire supérieur	Répertoire du milieu	Page	Déplacement
----------------------	-------------------------	-------------------------	------	-------------