



Système d'Exploitation 2

Chapitre 2 : La gestion de la mémoire

(partie 1)

1 ère Année Second Cycle

Dr. M. Baba Ahmed

Plan (partie 1)

- ▶ Introduction
- ▶ Rappels (hiérarchie des mémoires : mémoire physique, mémoire caches)
- ▶ Gestionnaire de la mémoire : objectifs de la gestion de la mémoire
- ▶ Espaces d'adressage d'un processus (Les différents types d'adresses, mappages d'adresses logiques et physiques, unité de gestion de la mémoire (MMU))
- ▶ La mémoire virtuelle (les problèmes pris en charge par la mémoire virtuelle, les avantages de la mémoire virtuelle, la réalisation de la mémoire virtuelle, permutation de programmes)

Introduction

- ❑ La mémoire est une ressource importante pour le stockage de programmes (et des données) en cours d'exécution
- ❑ Un programme ne peut s'exécuter que si ses instructions et ses données (au moins partiellement) sont en mémoire centrale.
- ❑ La multiprogrammation, pose comme condition obligatoire que la mémoire centrale soit partagée entre plusieurs processus.

Introduction

- ❑ Les capacités de stockage doublent en taille tous les 18 mois.
- ❑ Les programmes grossissent en taille **aussi vite que la mémoire**.
 - On aura toujours besoin de gestionnaires de mémoires performants.
- ❑ Le Gestionnaire de la mémoire gère l'hierarchie de mémoire (allouer, libérer, transfert de données, ex : mémoire physique et disque ...)
 - Mémoire cache
 - Mémoire centrale
 - Mémoire de masse (disques)

Objectifs de la gestion de la mémoire

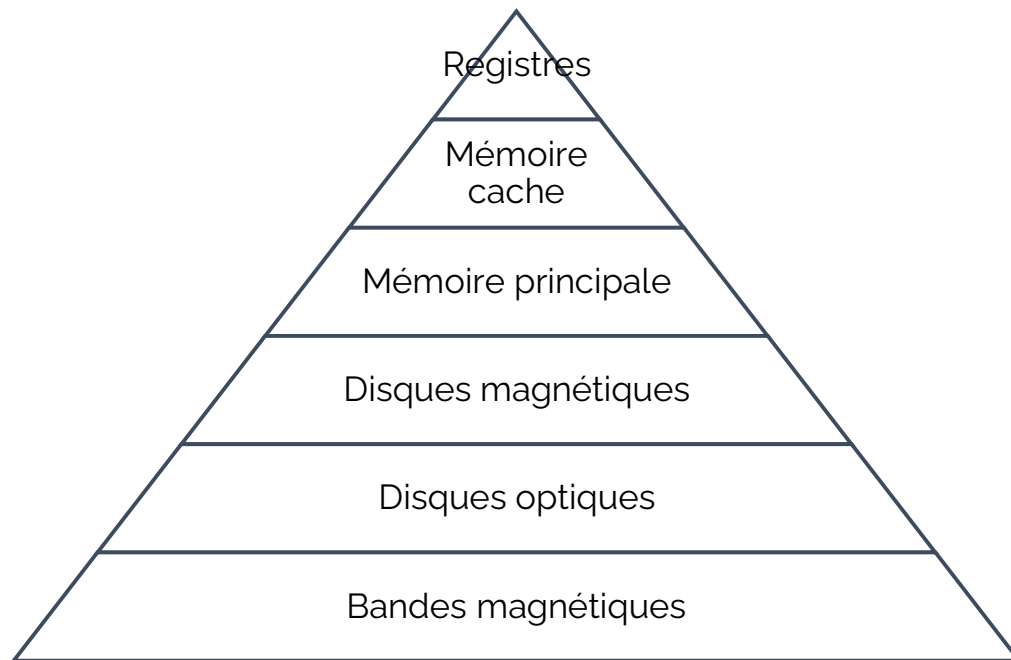
- ❑ Optimisation de l'utilisation de la mémoire principale RAM
- ❑ Le plus grand nombre possible de processus actifs doit y être gardé, de façon à optimiser le fonctionnement du système en multiprogrammation
 - Garder le système le plus occupé possible.
 - S'adapter aux besoins de mémoire de l'utilisateur, qui varient dans le temps: *allocation dynamique*
- ❑ La protection : La coexistence de plusieurs processus en mémoire nécessite la protection de chaque espace mémoire vis-à-vis des autres.
 - Un processus P1 ne peut accéder à l'espace d'un processus P2 que s'il est autorisé.

Rappel : Hiérarchie des mémoires

❑ Les différents éléments de la mémoire d'un ordinateur sont ordonnés en fonction des critères suivants :

- Temps d'accès
- Capacité
- Coût par bit (quantité

D'information échangé
par unité de temps)



Rappel : Hiérarchie des mémoires : Types d'accès

❑ Accès séquentiel : (exemple les bandes magnétiques)

- C'est l'accès le plus lent ; pour accéder à une information particulière on est obligé de parcourir toutes celles qui la précèdent

❑ Accès direct : (par exemple la mémoire centrale, les registres)

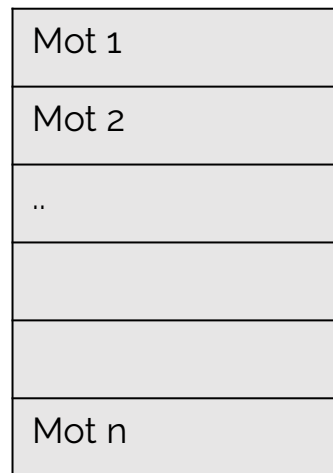
- les informations ont une adresse propre, et sont donc directement accessibles

❑ Accès semi-séquentiel :

- C'est une combinaison des accès direct et séquentiel.
- Dans un disque magnétique, l'accès au cylindre est direct et l'accès au secteur est séquentiel.

Rappel : Mémoire physique

- ❑ La mémoire physique est constituée d'un ensemble de mots **contigus** désignés chacun par une **adresse physique**.
 - Un mot mémoire est formé par un ou plusieurs octets.
- ❑ Le processeur accède aux mots de la mémoire centrale (MC) par le biais de deux registres:
 - Le registre **RAD** contient l'adresse du mot à lire ou l'adresse du mot où écrire,
 - le registre **RDO** contient soit la donnée à écrire soit la donnée lue.



Mémoire physique

Rappel : Mémoire cache

❑ Mémoire cache (vs RAM):

- Beaucoup plus couteuse, et d'une capacité de stockage plus petite
- Sa vitesse est beaucoup plus rapide (que la MC).

❑ La mémoire cache contient **les mots mémoire** les plus récemment accédés par le processeur.

- Lorsque le processeur demande à lire un mot, il le cherche avant dans le cache.
- S'il est présent, il y a succès et aucun accès à la MC sinon, sinon le mot ainsi que ses voisins sont chargés dans le cache (possibilité que les adresses mémoire voisines seront également accédées)

Espace d'adressage d'un processus

- ❑ L'espace d'adressage d'un processus est composé d'un ensemble de mots (adressables), il est composé de 4 parties:

Code
Données
Pile (stack)
Tas (heap)

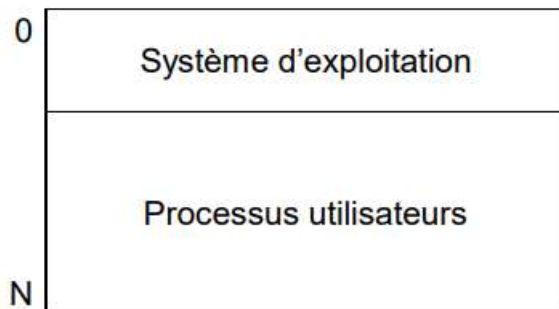
- ❑ Cet ensemble d'adresse est appelé espace d'adresse **logiques** ou **virtuelles**
Remarque : Tas (heap) pour les allocations dynamiques (ex : pointeurs)

Différents types d'adresses

- ❑ L'adresse logique est générée (vu) par le CPU pendant l'exécution d'un programme
- ❑ L'adresse physique fait référence à un emplacement dans l'unité de mémoire.
 - vue par l'unité de mémoire
- ❑ L'ensemble de toutes les adresses logiques générées (vu) par le CPU en référence à un programme est appelé espace d'adressage logique.
- ❑ L'ensemble de toutes les adresses physiques mappées aux adresses logiques correspondantes est appelé adresse physique.

Partition de la mémoire centrale

- ❑ La mémoire principale peut loger le SE et les différents processus utilisateurs. La mémoire est habituellement subdivisée en deux partitions : une pour le SE résident et l'autre pour les processus utilisateurs



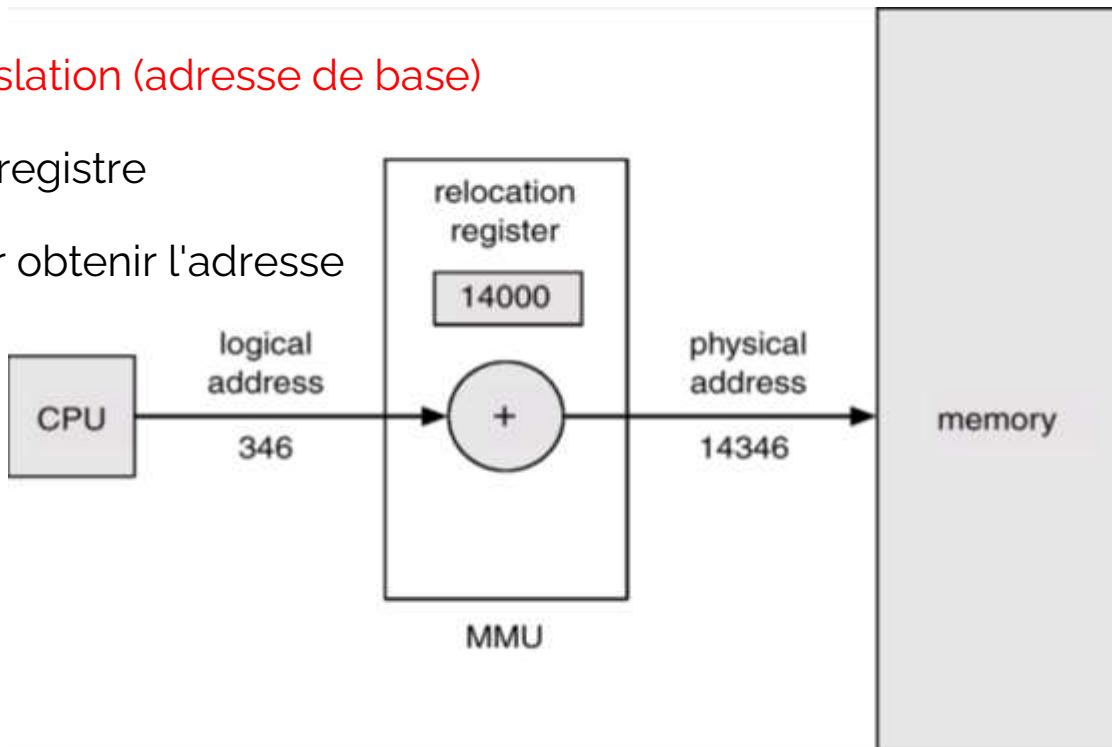
Mappage d'adresses logiques et physiques

- ❑ L'ensemble des adresses physiques réellement occupé par le programme exécutable suite à son chargement en MC est appelé **espace d'adresses physiques**.
 - On dit alors qu'il y a un mappage entre l'espace d'adresses logiques et l'espace d'adresses physiques.
- ❑ Pour que le processeur puisse accéder à la MC, il est nécessaire de convertir les adresses logiques générées par le processeur en adresses physiques.
 - Cette conversion est réalisée par un dispositif matériel appelé la **MMU (Memory Management Unit)**.

Mappage d'adresses logiques et physiques

❑ La valeur du registre de translation (adresse de base)

il consiste à ajouter la valeur du registre de base à l'adresse logique pour obtenir l'adresse physique correspondante



Unité de gestion de mémoire (MMU)

❑ **Memory Management Unit**

- Matériel de mapping des adresses virtuelles en adresses physiques

❑ La valeur du registre d'emplacement est ajoutée à chaque adresse générée par un processus (l'espace logique)

- Le processus traite des adresses logiques, il ne voit jamais les adresses physiques

❑ **Remarque:** Les adresses physiques n'ont pas besoin d'être contigus

Principe

- ❑ Une adresse logique est une adresse dans un programme
 - Par rapport au programme lui-même seulement
 - Indépendante de la position du programme en mémoire physique
 - Les programmes ne connaissent que des adresses logiques.
 - C'est l'unité mémoire qui manipule les adresses physiques (emplacement mémoire)

La mémoire virtuelle

- Lorsqu'un programme est en exécution, il n'est pas nécessaire qu'il soit entièrement dans la mémoire principale ! Seules les parties qui sont en cours d'exécution ont besoin d'être dans la mémoire centrale
- Les autres parties peuvent être dans **la mémoire secondaire** (ex. disque), prêtes à être amenées en mémoire centrale sur demande
- Ceci rend possible l'exécution de programmes beaucoup plus grands que la mémoire physique
- Technique **indispensable** intégrée aujourd'hui dans la plus part des SE

Problème pris en charge par la mémoire virtuelle

❑ Problème 1: Capacité de la mémoire

- Taille d'un programme > taille de la MC
- Σ tailles des programmes en cours > taille de la MC

❑ Problème 2: Cohabitation entre plusieurs processus:

- Problème d'adressage (ou chaque processus doit disposer de son propre espace d'adressage et s'exécuter indépendamment des autres)
- Fragmentation de la mémoire (un processus doit s'apercevoir que son espace mémoire est continu, même si physiquement il est fragmenté)
- Protection des processus (contrôle d'accès à la mémoire, isolation).

Avantage de la mémoire virtuelle

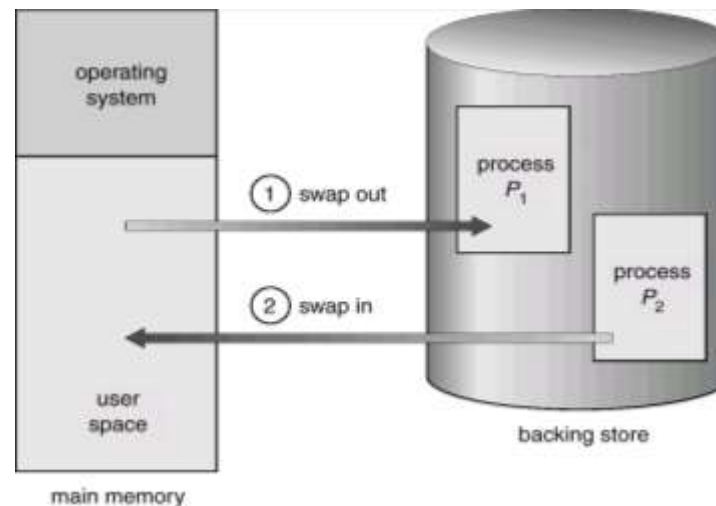
- ❑ Avoir plus d'espace adressable
 - La taille d'un programme n'est plus limitée par la taille de la mémoire physique.
 - Permet d'exécuter plus de programmes en même temps.
 - Plus de processus peuvent être maintenus en exécution en mémoire
- ❑ Protection mémoire entre processus

Réalisation de la mémoire virtuelle

- ❑ 3 Mécanismes sont utilisés pour la réalisation de la mémoire virtuelle:
 - Swapping (va-et-vient)
 - Pagination
 - Segmentation
 - Segmentation paginée

Permutation de programme (swapping)

- ❑ Consiste à transférer les blocs mémoire de la mémoire principale vers la mémoire secondaire ou vice-versa (va-et-vient).
 - Le va-et-vient est mis en œuvre lorsque tous les processus **ne peuvent pas tenir simultanément en mémoire.**
- ❑ Processus ou portion de Processus permuté sur disque pour sauver la mémoire quand ils ne sont pas actifs



Permutation de programmes (swapping)

- ❑ Un processus qui est inactif (soit bloqué, soit à l'état prêt)
 - Peut être déplacé temporairement sur une partie réservée du disque, appelée **mémoire de réserve** (**Swap Area** ou **Backing Store**).
 - Cette opération est connue sous le nom de **Swap-Out**.
- ❑ Le processus déplacé sur le disque sera ultérieurement rechargé en mémoire pour lui permettre de poursuivre son exécution ;
 - On parle dans ce cas d'une opération **Swap-In**.