Allocation de mémoire

SYE
9 - Allocation de mémoire

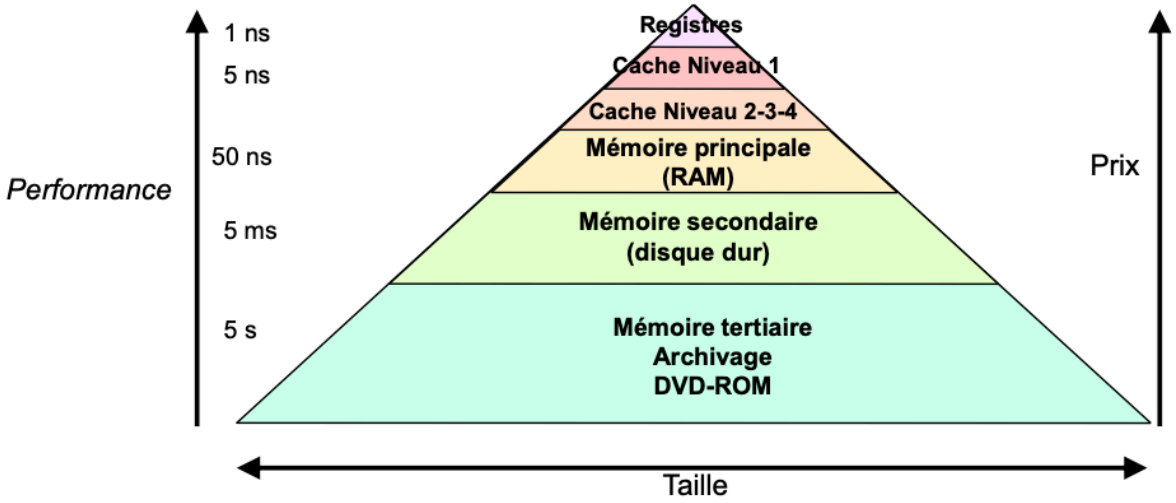
Résumé du document
Definition

Table des matières

1. Dispositif mémoire

Dans un système informatique, la mémoire est un dispositif qui permet de stocker des données et des instructions. Elle est divisée en trois catégories :

- Les registres : mémoire interne du processeur
- La mémoire RAM : mémoire vive
- La mémoire de masse : disque dur, clé USB, etc.

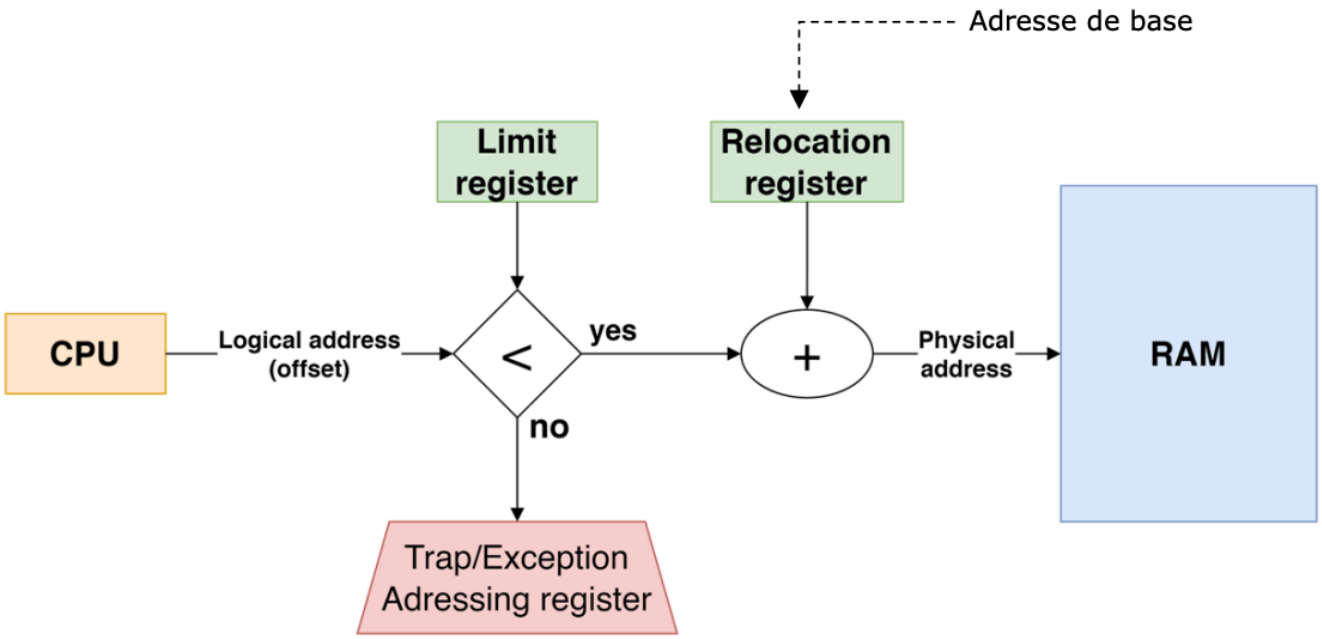


2. Allocation de la mémoire physique

Le chargeur (**loader**) d'une image binaire doit trouver un emplacement disponible en mémoire pour placer l'image.

- Détermination de l'adresse de base
  - Toutes les adresses du code binaire sont réajustées pour maintenir la validité des sauts et références.
  - En système **monoprogrammé**, l'image binaire contient des adresses absolues, donc aucun réajustement n'est nécessaire.
  - En système **multiprogrammé**, un réajustement est indispensable.

L'adresse de base d'un processus indique le début de son emplacement en RAM.

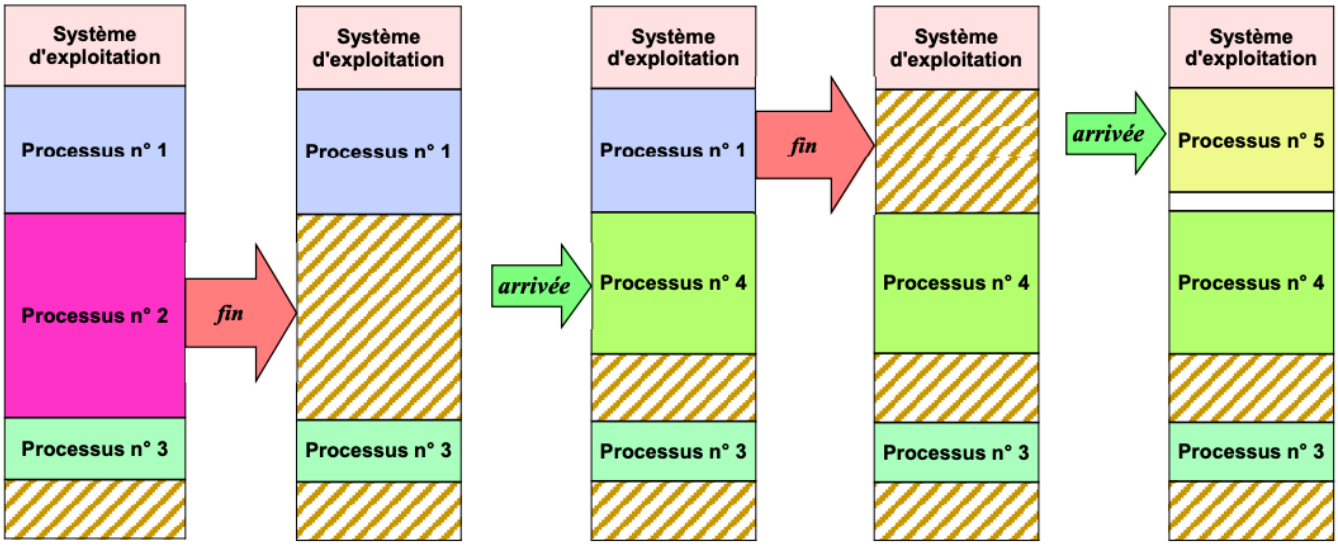


- Lorsque le chargeur détermine cette adresse :
  - Il configure un registre de "relocation" pour ajouter cette adresse aux adresses générées par le CPU.
  - Ce mécanisme évite la modification de l'image binaire et nécessite un support matériel (registre dédié).
  - Sans support matériel, le chargeur effectue la conversion des adresses relatives en absolues.

2.1. Fragmentation

Le mouvement des processus (va-et-vient entre le disque et la mémoire) génère une fragmentation au fil du temps. Au bout d'un certain temps, la mémoire se trouve très fragmentée et il devient difficile de trouver un emplacement disponible pour charger un processus, alors que la somme des trous mémoire constituerait une taille mémoire suffisamment grand.

- Fragmentation mémoire au fil du temps.
- Difficulté à relocaliser le code.
- Complexité dans la gestion dynamique des adresses.



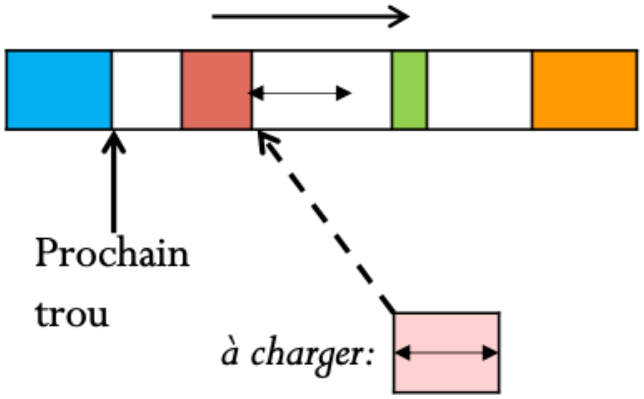
La fragmentation mémoire complique l'allocation de nouveaux processus, bien que l'espace total puisse suffire.

- **Solution** : Défragmentation mémoire, mais celle-ci
  - Nécessite le réajustement des adresses et instructions.
  - Est complexe et lente.
- **Alternative moderne** : Utilisation de registres de base dans les processeurs
  - Les adresses sont stockées comme des **offsets** relatifs à l'adresse de base.
  - Déplacer un processus revient à modifier l'adresse de base dans le registre.

3. Algorithme d'allocation

3.1. First-Fit

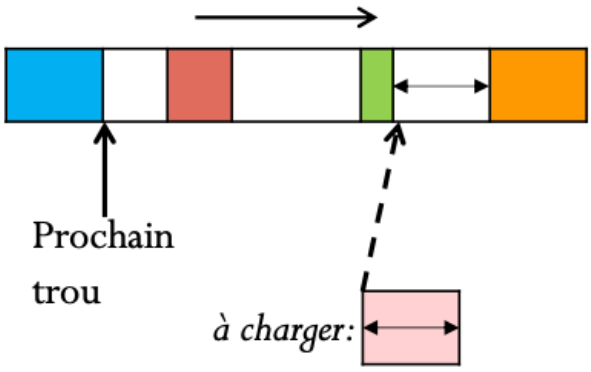
Cet algorithme recherche la première zone libre satisfaisante. Il est simple et rapide, mais peut générer une fragmentation importante.



3.2. Best-Fit

Cet algorithme recherche la zone libre la plus petite qui peut contenir le processus. Il est plus lent que **First-Fit** mais génère moins de fragmentation. Il a pour ojectif de générer un résidu minimal.

- Le point faible de cet algorithme est qu'il doit parcrourir **toute la mémoire** pour chaque allocation.



3.3. Quick-Fit

L'allocation Quick-Fit est une amélioration de l'algorithme **Best-Fit**. Elle divise la mémoire en plusieurs listes de tailles prédéfinies. Chaque liste contient des blocs de mémoire de la même taille. On parle donc d'une liste à 2 dimensions : la première dimension est la **taille des blocs**, la seconde dimension est la **liste des blocs** de cette taille.

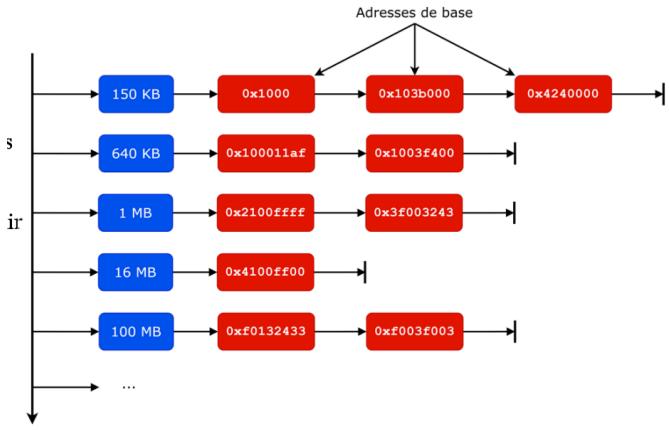
Ce type d'allocation permet:
- Une fusion des zones adjacentes
- Une sélection rapide de la zone libre
- Des résidus minimaux

3.3.1. Complément du cours

*Chaque requête d'allocation ou de retrait exige une **mise à jour de la liste**. Lorsque la mémoire est mise à forte contribution, une telle liste peut contenir beaucoup d'entrées. C'est pourquoi, il est aussi nécessaire, périodiquement, de fusionner des trous adjacents en mettant à jour la liste. Cette opération pouvant demander un certain temps CPU, on peut l'imaginer qu'elle s'effectue en arrière-plan (background).*

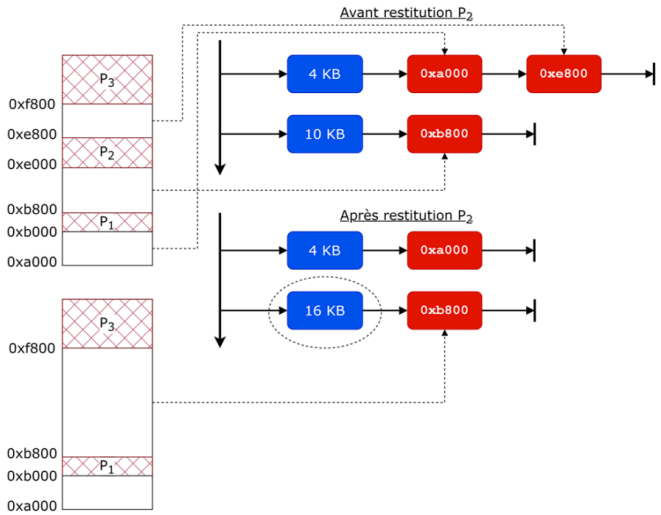
3.3.2. Gestion de la liste

On visualise donc sur cette image la structure de la liste de blocs de mémoire. Chaque liste est composée de blocs de mémoire de même taille. Lorsqu'un bloc est alloué, il est retiré de la liste. Lorsqu'un bloc est libéré, il est ajouté à la liste.



3.3.3. Restitution

Dans cette illustration nous pouvons voir comment se passe le processus de fusion de deux blocs adjacents. Lorsqu'un bloc est libéré, il est ajouté à la liste. Si un bloc adjacent est également libre, les deux blocs sont fusionnés pour former un bloc plus grand.



4. Espace d'adressage virtuel

L'adressage virutel consiste à traduire les adresses virtuelles en adresses physiques.

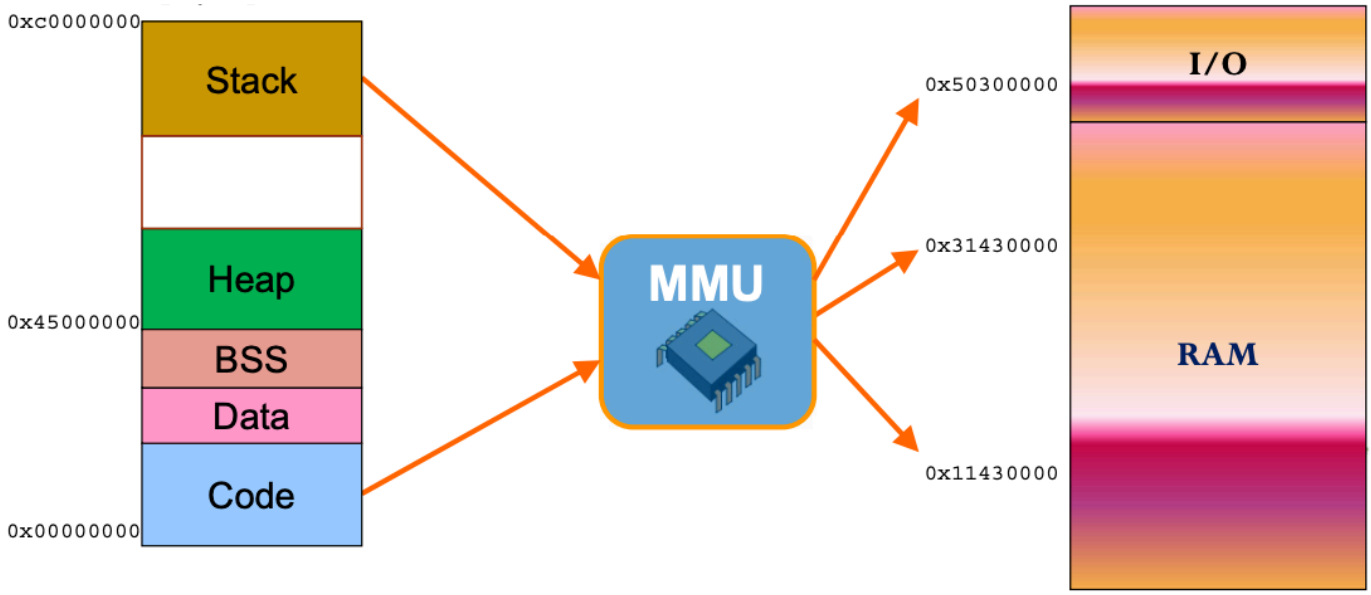
L'idée de la **réimplantation dynamique** est de donner à tous les processus une traduction différente. Cette traduction repose sur l'utilisation de table et nécessistera une reconfiguration matérielle à chauqe changement de processus.

Complément

- *Il faut noter que toutes les adresses virtuelles que constitue l'espace d'adressage virtuel n'ont pas forcément une équivalence en adresses physiques. Généralement, seules les adresses virtuelles "utiles" trouveront une traduction vers une adresse physique. Ainsi, un espace d'adressage virtuel peut se révéler beaucoup plus grand que l'espace d'adressage physique, dans sa représentation, mais en réalité seul un sous-ensemble de ses adresses trouveront une équivalence physique.*

4.1. MMU (Memory Management Unit)

La MMU est un composant matériel qui permet de traduire les adresses virtuelles en adresses physiques. Les processeurs ne possèdant pas de MMU ne peuvent pas gérer l'adressage virtuel. Un changement de contete nécessitera systèmatiquement un changement de traduction des adresses virtuelles. De ce fait la MMU devra être reconfigurée à chaque changement de processus.



4.2. Partitionnement de la mémoire

L'espace d'adressage correspond à l'ensemble des adresses visibles par un processus. Sur une architecture 32 bits, cet espace, qui peut aller jusqu'à 4 Go, est divisé en deux parties : une partie utilisateur et une partie noyau. La partie utilisateur, généralement fixée à 3 Go, contient les données du processus telles que le code, les données et la pile. Le processeur fonctionne alors en mode utilisateur. La partie noyau, occupant généralement 1 Go, est réservée au système d'exploitation et à ses structures internes. Le noyau s'exécute en mode noyau, avec un accès complet à toutes les instructions et adresses mémoire.

D'autres configurations de partitionnement existent, comme un découpage 2G/2G ou même 4G/4G. Dans ce dernier cas, une reconfiguration de la MMU (Memory Management Unit) est nécessaire non seulement lors d'un changement de contexte entre processus, mais aussi lors d'une transition entre les modes utilisateur et noyau. Bien que cette configuration ait été proposée dans le noyau Linux, le découpage par défaut reste 3G/1G.

L'allocation d'un espace complet de 4 Go par processus n'est possible que si la mémoire est virtualisée grâce à une MMU. Sans cette unité, l'espace d'adressage se limite à la taille de la RAM physique disponible.