

# Introduction

La mémoire a toujours été une ressource indispensable d'un système. Elle est partagée entre tous les processus. La mémoire est une suite non structurée d'octets, le système d'exploitation ne connaît donc pas la structure des informations en mémoire (qui dépendent de chaque processus).

## Importance de la mémoire

Les instructions d'un programme peuvent contenir des adresses mémoires en argument, il est ainsi nécessaire qu'un processus se trouve entièrement en mémoire pour pouvoir s'exécuter. Une gestion efficace de cette mémoire est alors primordiale.

## Types de mémoires

### Registres

Les registres sont des zones mémoires attachées au processus, leur taille est généralement très réduite (de l'ordre de quelques Ko) mais sont extrêmement rapides, pouvant généralement être accédé en un seul cycle d'horloge du processeur.

### Mémoire vive (RAM)

La mémoire vive est la mémoire principale du système. Elle est la ressource importante à gérer et est souvent présente en quantité (entre deux et 64 Go). Elle est standardisée aux normes DDR3 et DDR4. Elle est plus lente par rapport au CPU (il faut parfois plusieurs cycles d'horloge pour récupérer une information), le CPU doit donc attendre ou mettre la donnée en cache. Cependant, cette ressource reste beaucoup plus rapide qu'un disque (SSD ou HDD).

### Mémoire cache

La mémoire cache a pour but d'améliorer les performances générales du système en planquant en mémoire plus rapide les données fréquemment demandées.

Il existe trois niveaux de mémoire cache, le premier niveau sert à stocker temporairement des instructions et données. Les deux autres niveaux sont présents entre la mémoire RAM et la mémoire cache de premier niveau.

# Mémoire virtuelle

La mémoire virtuelle est une mémoire simulée par le système. Elle utilise le disque dur comme RAM additionnelle. Elle est seulement limitée par la taille du disque dur, mais elle est très lente.

## Isolation de la mémoire

Il est essentiel que chaque processus ait une zone mémoire réservée pour éviter des problèmes de sécurité et d'intégrité des données.

Un processus ne peut donc pas écrire dans l'espace mémoire d'un autre. S'il essaye, on aura une "Segmentation Fault". La mémoire n'est de ce fait pas partagée entre les processus sauf lorsque cela est explicitement demandé. (Voir [Mémoire partagée](#) pour plus d'informations).

## Translation d'adresse

Lorsqu'un programme est chargé, il est placé entièrement en mémoire à une adresse de départ. Son espace d'adressage est défini et les instructions du programme font référence à l'adresse 0.

La translation peut être effectuée à 3 moments :

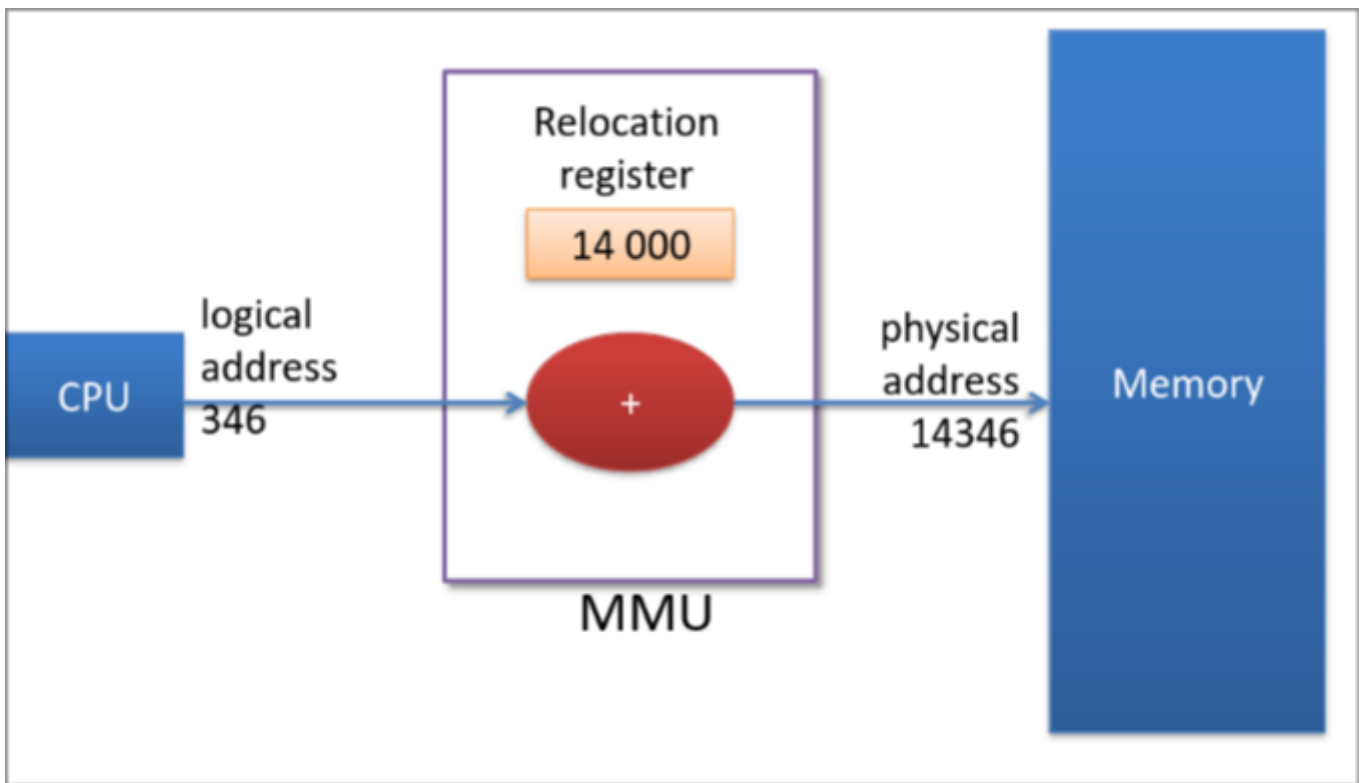
- À la **compilation** (inexistant aujourd'hui), on met les adresses physiques dans le programme compilé ;
- Au **chargement** (l'adresse de départ est choisie lors du chargement du processus par le système d'exploitation) ;
- A l'**exécution**, lors de l'exécution des instructions, cela demande un hardware précis et est utilisé par la segmentation.

## Types d'adresses

On parle d'**adresse logique** pour parler de l'adresse présente dans les programmes (qui se réfèrent à l'adresse 0).

On parle en revanche d'**adresse physique** pour parler d'une case mémoire adressable de la mémoire RAM.

La conversion entre l'adresse **logique** et l'adresse **physique** est faite par le **Memory Management Unit (MMU)** qui est un composant matériel spécialisé dans l'opération de translation.



Revision #3

Created 3 January 2024 08:56:35 by SnowCode

Updated 6 January 2024 19:13:15 by SnowCode