

# Segmentation et pagination

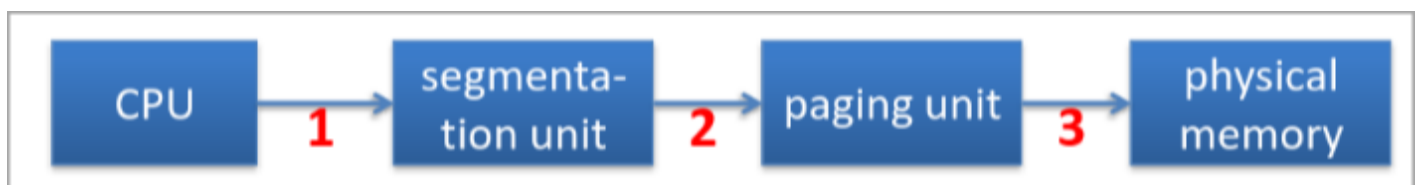
Pour combiner les avantages de la [segmentation](#) avec les avantages de la [pagination](#), on va regarder à l'exemple du fonctionnement de l'architecture des processeurs Intel 32 bits.

Ce processeur peut gérer un maximum de 16 384 segments de maximum 4 Go. L'espace d'adressage est découpé en deux parts égales. Une partition pour les segments privés (par exemple, processus) et une autre pour les segments partagés (par exemple, librairie partagée).

Il y a également deux tables, la LDT (Logical Descriptor Table) pour la partition des segments privés, et la GDT (Global Descriptor Table) pour les segments partagés.

## Conversion vers adresse physique

Pour passer de l'adresse logique à l'adresse physique, il va falloir passer par deux intermédiaires :



- L'**unité de segmentation** qui transforme l'adresse logique en adresse linéaire
- L'**unité de pagination** qui transforme l'adresse linéaire en adresse physique.

## Unité de segmentation

L'adresse logique est composée d'un sélecteur et d'un déplacement. Le sélecteur est lui-même composé de 3 informations :

- Le numéro du segment (noté *s*)
- Si c'est un segment privé ou partagé (LDT ou GDT) (noté *g*)
- Des informations de protection (noté *p*)

À partir de cette adresse logique, l'unité de segmentation construit l'adresse linéaire.

## Unité de pagination

L'adresse linéaire est composée de 3 informations,

- Le **directory**, qui indique la table des pages à utiliser parmi le répertoire des tables de pages du processus courant

- La **page**, qui indique la page dans la table des pages
- Le **offset** (déplacement), qui est le même que celui cité dans l'adresse logique.

À partir de cette adresse linéaire, l'unité de pagination construit l'adresse physique comme vu précédemment dans le chapitre sur la [Pagination](#).

---

Revision #1

Created 3 January 2024 14:40:11 by SnowCode

Updated 6 January 2024 19:13:15 by SnowCode