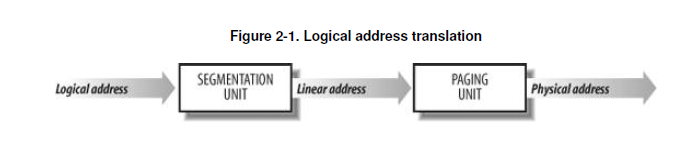
MMU fait la translation entre adresses physiques et adresses logiques (mémoire virtuelle)

Page (mémoire virt) et frames (mémoire physique) de 4 ou 8 kb en RAM

Une partie de la RAM fait office de cache

Une partie de la RAM stocke l’image du noyau et ses structures

Le reste



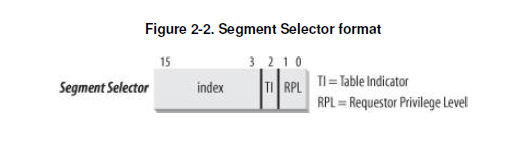
Segments : taille différentes et pages : taille fixes

RAM découpée en frames de 4k

Memory arbiter garantit les accès concurrents à la RAM (plusieurs CPU, DMA…)

Segmentation :

16 bits pour le segment identifier (ou segment selector) et 32 pour l’offset dans le segment



CS : codesegment register

SS : stack segment register

DS : data segment register

Pagination :

Gère les droits.

Table des pages gère la correspondance

Physical memory

* Nodes : bande de mémoire référencé par pg\_data\_t (struct pglist\_data) dans linux/mmzone.h. Tous les nodes sont référencé dans une pgdat\_list . La liste est chainée le next est situé dans pg\_data\_t
* Un node est divisé en zones référencé par zone\_t (struct zone\_struct) dans linux/mmzone.h
  + ZONE\_DMA : @basses : first 16 Mb
  + ZONE\_NORMAL : 16 -> 896 Mb
  + ZONE\_HIGHMEM -> >896 Mb
* Page frame référencé par struct page et toutes les pages dans un tableau mem\_map