

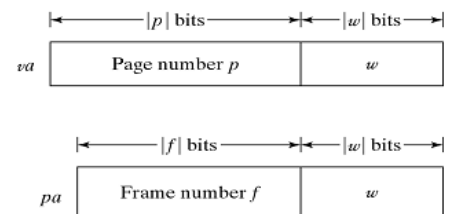
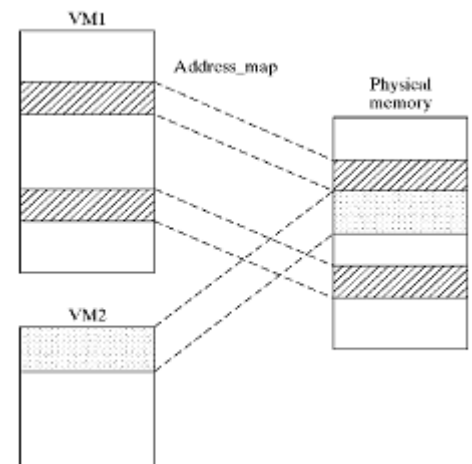
# 7. Lageradministration - Virtuelt lager

## Paging

1. Fast størrelse den virtuelle hukommelse deles ind i
2.  $|\text{Page}| = |\text{PageFrame (i det fysiske lager)}|$
3. En page indeholder words som kan tilgås med parret  $(p, w)$  //  $(\text{page}/\text{word})$
4. Holdes styr på i frame table eller page table.

## Adresseoversættelse → pagetabel

1. **Address maps:** at få en fysisk plads i hukommelsen fra en logisk adresse.
2. Frame table: En for hele systemet. Holder styr på page og pid på den proces siden tilhører.
  - a. Kan tage lang tid at kigge det hele igennem (kan bruge associativ memory (translation look-aside buffers – form for cache))
3. **Page table:** Ligger i RAM – PTR register, da vi ellers skal kigge i rammen efter "offsettet"
  - a. Bruger meget plads



## Page faults

1. **Page Fault:** Et program skal bruge en page fra rammen, som ikke er der. Så skal den hentes ind fra harddisken.
2. **Global strategi:** antager at der er et fast antal page frames til deling mellem alle processer.
  - a. FIFO, LRU (least recently used), Second-chance
  - b. Second-chance: En bit med hver page sat til 1. Når der er page fault løbes alle bit igennem indtil man finder et 0. Den side med 0 som bit skiftes ud. Når en bit med værdien 1 læses sættes værdien til 0. Hver gang en side i hukommelsen læses, sættes dens tilhørende bit til 1.
3. **Lokal strategi (ser kun på de pages der er allokeret til processen).**
  - a. Eksempel: Page Fault Frequency Replacement Algorithm (PFF) - fjerner automatisk pages der ikke er blevet læst indenfor et vist tidsrum. OS'et sørger for at lave page faults med et bestemt interval. Ved hver page faults, fjernes alle sider der ikke er blevet læst, siden sidste page fault. Dette giver 5 page faults.

