Lecture 2 — review 2

1. review of pipelining & caches

2. virtual memory   "virtual" ≠ 不存在
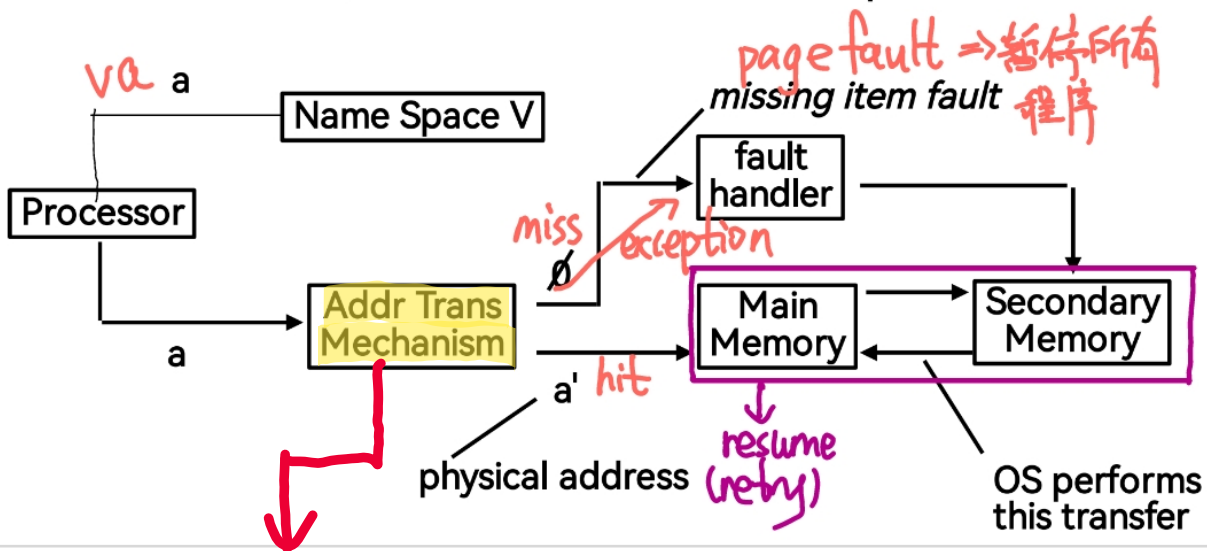
secondary memory (更大的"内存")
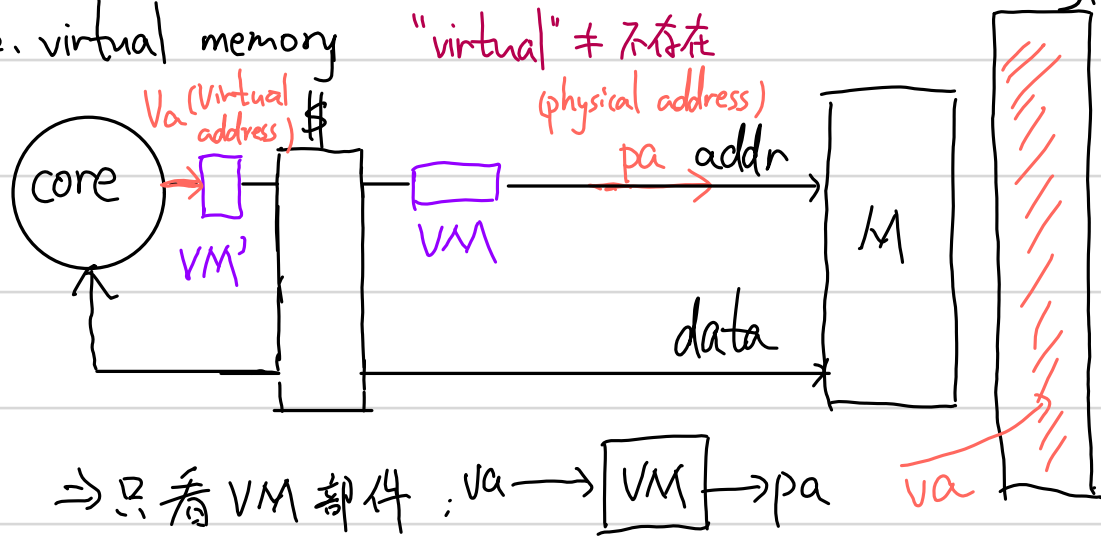
⇒ 只看 VM 部件: va ⟶ [VM] ⟹ pa

若无翻译部件 (VM): va = pa
若有 VM: va ≠ pa
VM 在 $前/后: cache 得到 pa/va
⇒ 两种不同的 cache

Secondary Memory 可能为: disk, 云(网络)... ⇒ 此时 Mem 相当于它的一个 cache

page fault ⇒ 暂停所有程序
missing item fault

OS performs this transfer
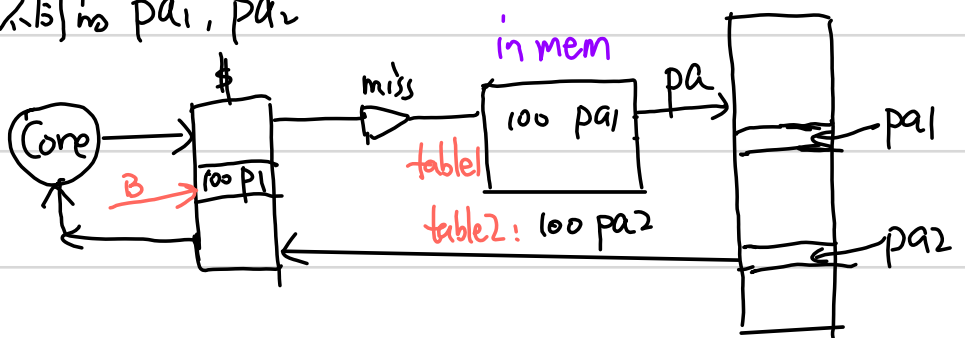
Address Mapping

actually, conca is more likely

高低位拼接
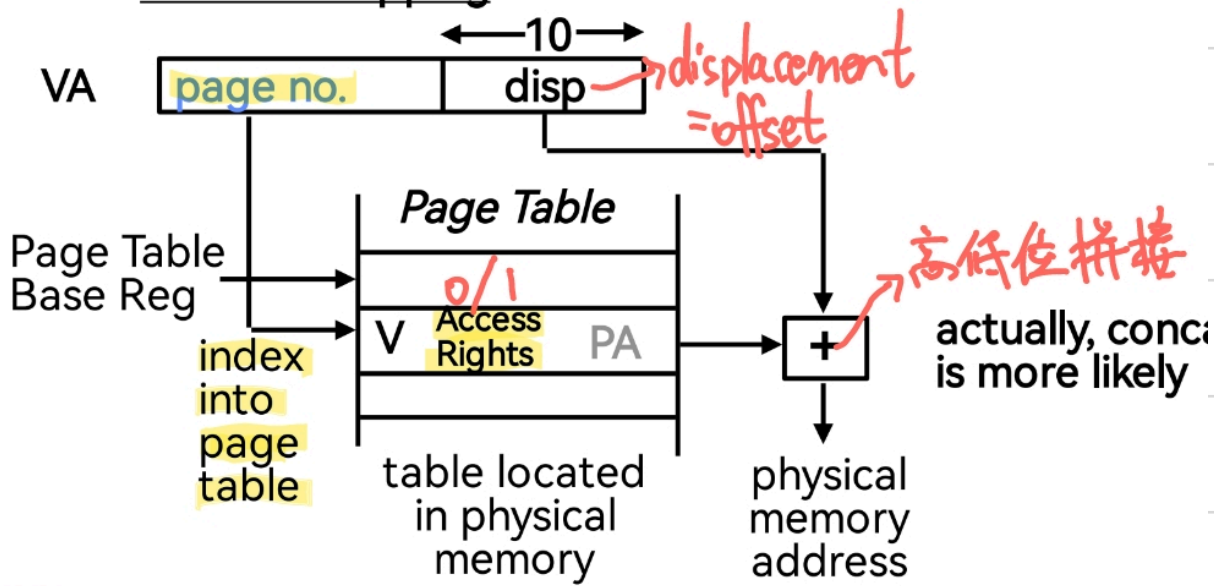
1/19/01

A. VM 放在 $ 前 or 后? 若放在 $后 ⇒ VA cache has problem!
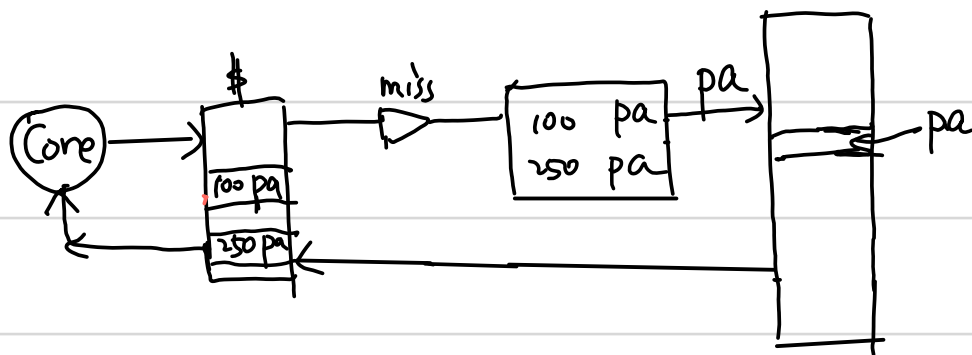
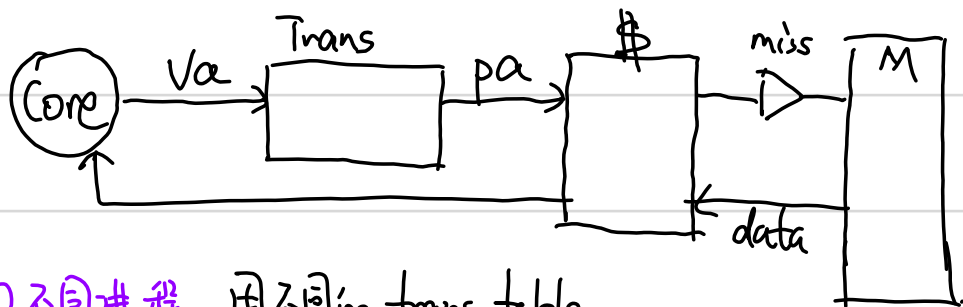① synonym: 2个不同进程 A,B, 同一个 va 对应不同的 pa₁, pa₂

在进程切换时, B 可能会取到错的 pa₁

(不同进程用不同页表$)

② alias 别名: 用2种不同方式取数据(打开文件 & block)时, 可能导致不同 va 对应同一 pa

⇒ 在 cache 中会出现同一内存有2处位置(有别名), 如果修改了其中一处, 则出现 inconsistency

B. 如果 VM 放在 $ 前: PA cache: no such problems

总而言之, cache 只与 pa 相关, 不关心 va
⇒ 不出问题

① 不同进程 用不同的 trans table
但映到同一个 pa, cache 中无问题
or 不同的

② 不同 va → 同一 pa cache 中始终是 pa
alias 不会有别名

但: PA cache 导致 performance 下降: 即使 cache hit 也要先在 trans table 里翻译, 而 table 在 mem 中, 故时间长

⇒ 想办法解决 VA cache 的问题: