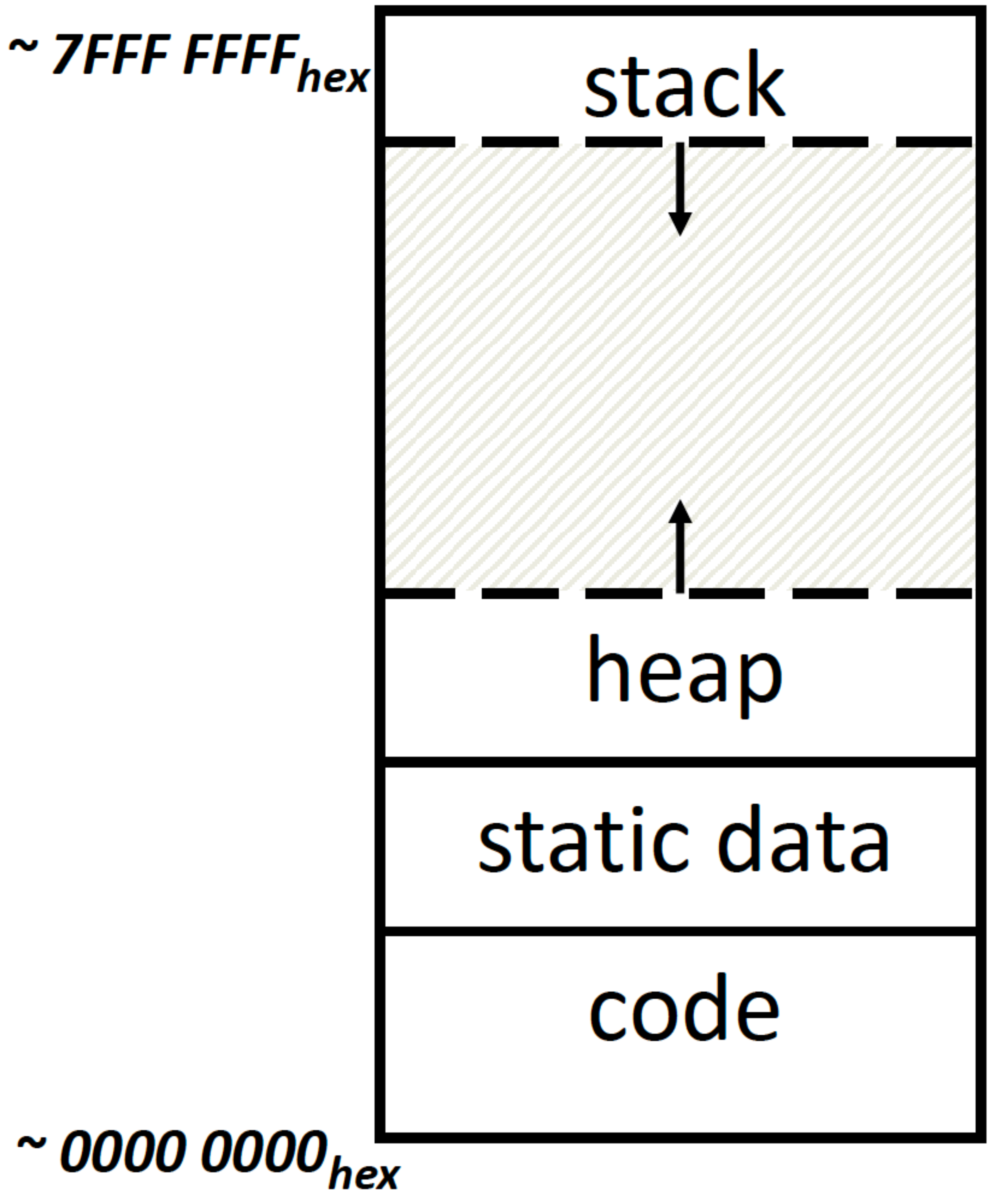
CA DISCUSSION13

VIRTUAL MEMORY

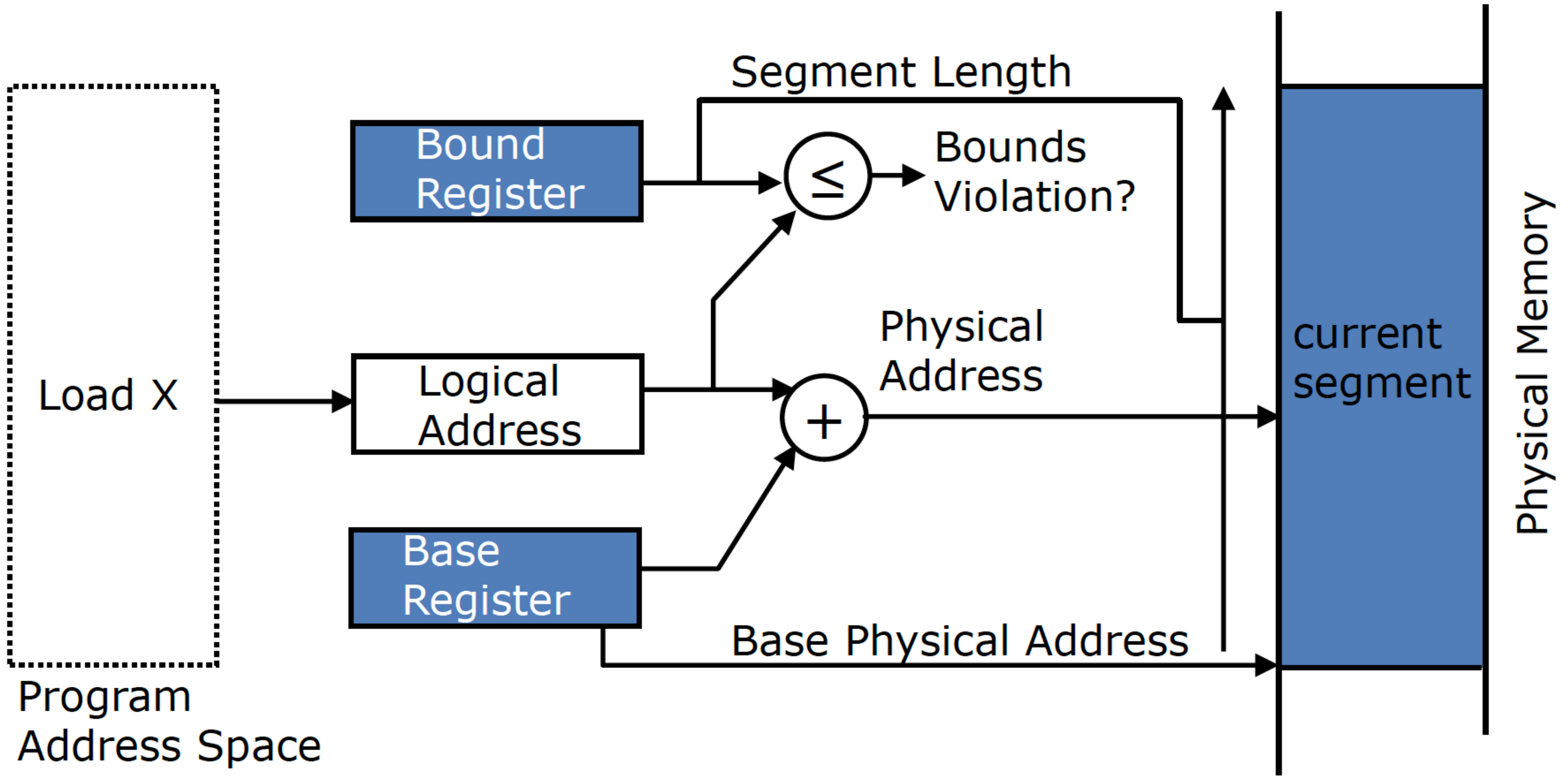
2021.5.24 彭绍庭

WHY DO WE NEED VM

- Adding disk to hierarchy
- Give apps a virtual view of memory
- Protection between processes



Base and bound



But: Memory fragmentation

Virtual memory

In computing, **virtual memory**, or **virtual storage**[b] is a memory management technique that provides an "idealized abstraction of the storage resources that are actually available on a given machine"[3] which "creates the illusion to users of a very large (main) memory".[4]

Virtual Address (VA) What your program uses

Virtual Page Number (VPN)	Page Offset

Physical Address (PA) What actually determines where in memory to go

Physical Page Number (PPN)	Page Offset

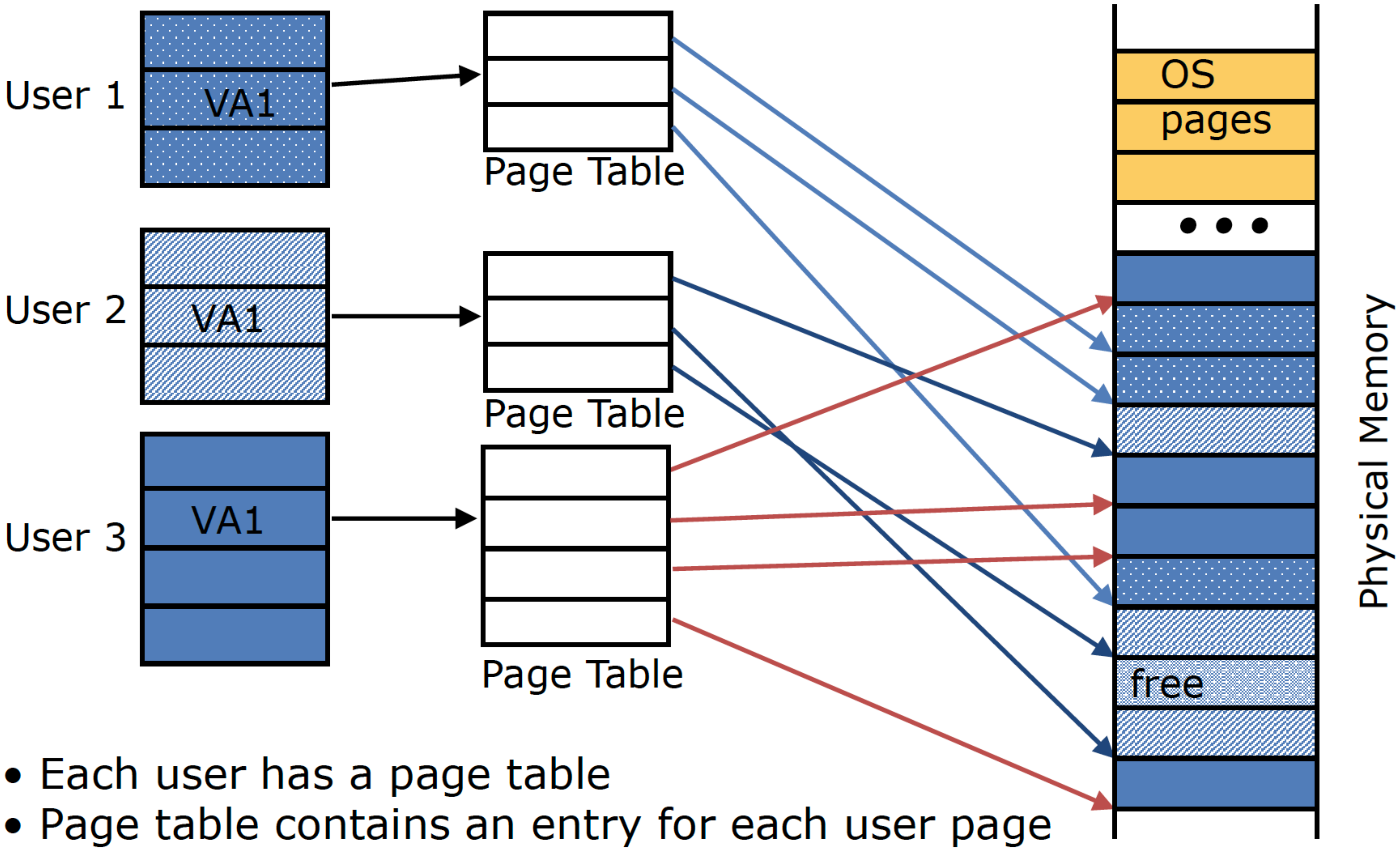
Page and page table

In VM, we deal with individual *pages*

- Usually ~4 KB on modern systems
  - Larger sizes also available: 4MB, very modern 1GB!
- Now, we'll "divide" memory into a set of pages

Valid	Dirty	Permission Bits	PPN
— *Page entry (VPN: 0)* —			
— *Page entry (VPN: 1)* —			

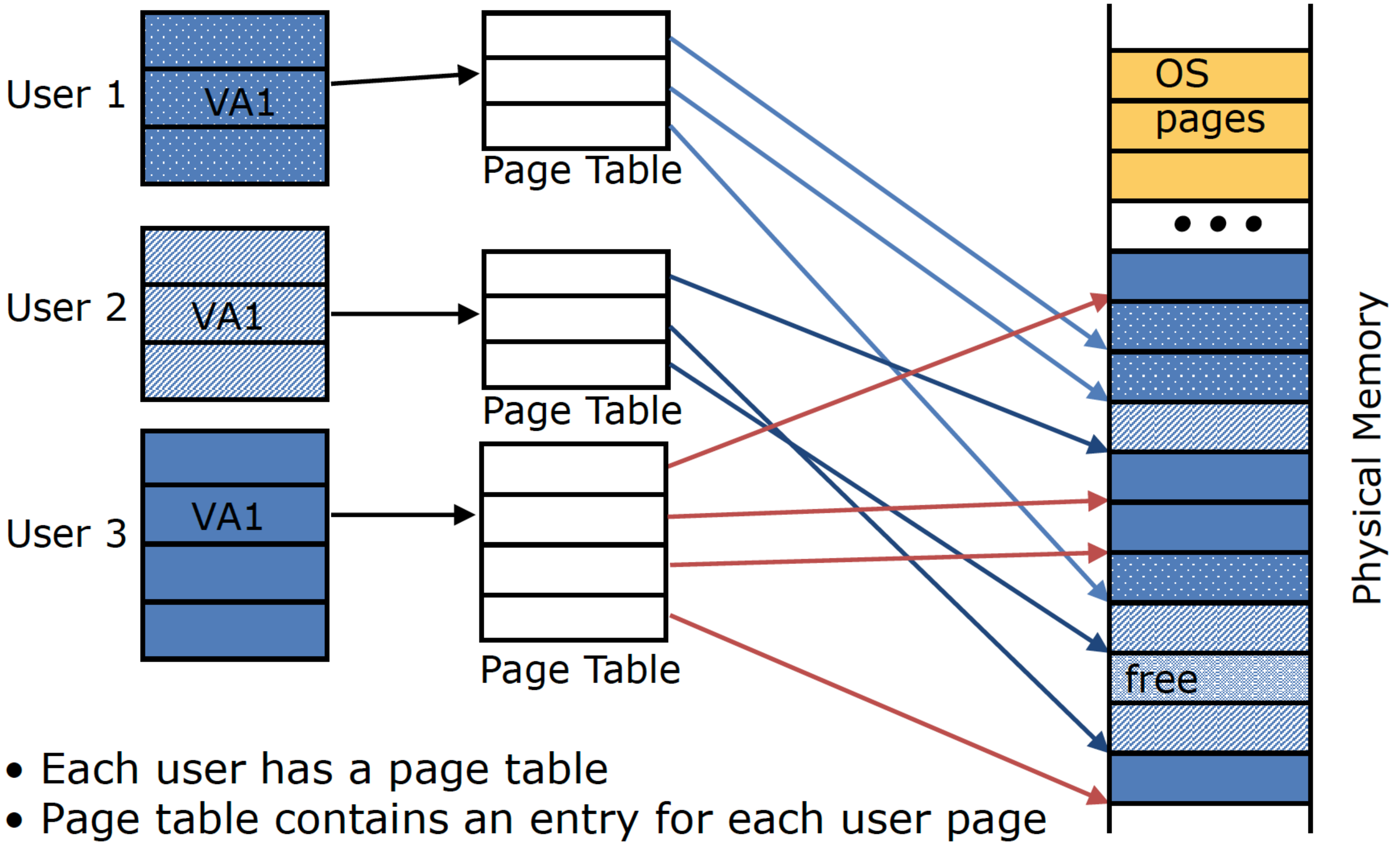
Page and page table



- Each user has a page table
- Page table contains an entry for each user page

Page fault

A **page fault** (sometimes called **#PF**, **PF** or **hard fault**)[a] is a type of exception raised by computer hardware when a running program accesses a memory page that is not currently mapped by the memory management unit (MMU) into the virtual address space of a process.



- Each user has a page table
- Page table contains an entry for each user page

Swapping

- When physical memory is used up, “evict” a page from physical memory to disk (swap out).
- When programs need this evicted page, a “page fault” will be generated, and OS will swap it into memory (may cause the eviction of another page).
- How to decide which page to evict?
- FIFO
  Clock algorithm
  Second chance algorithm ...

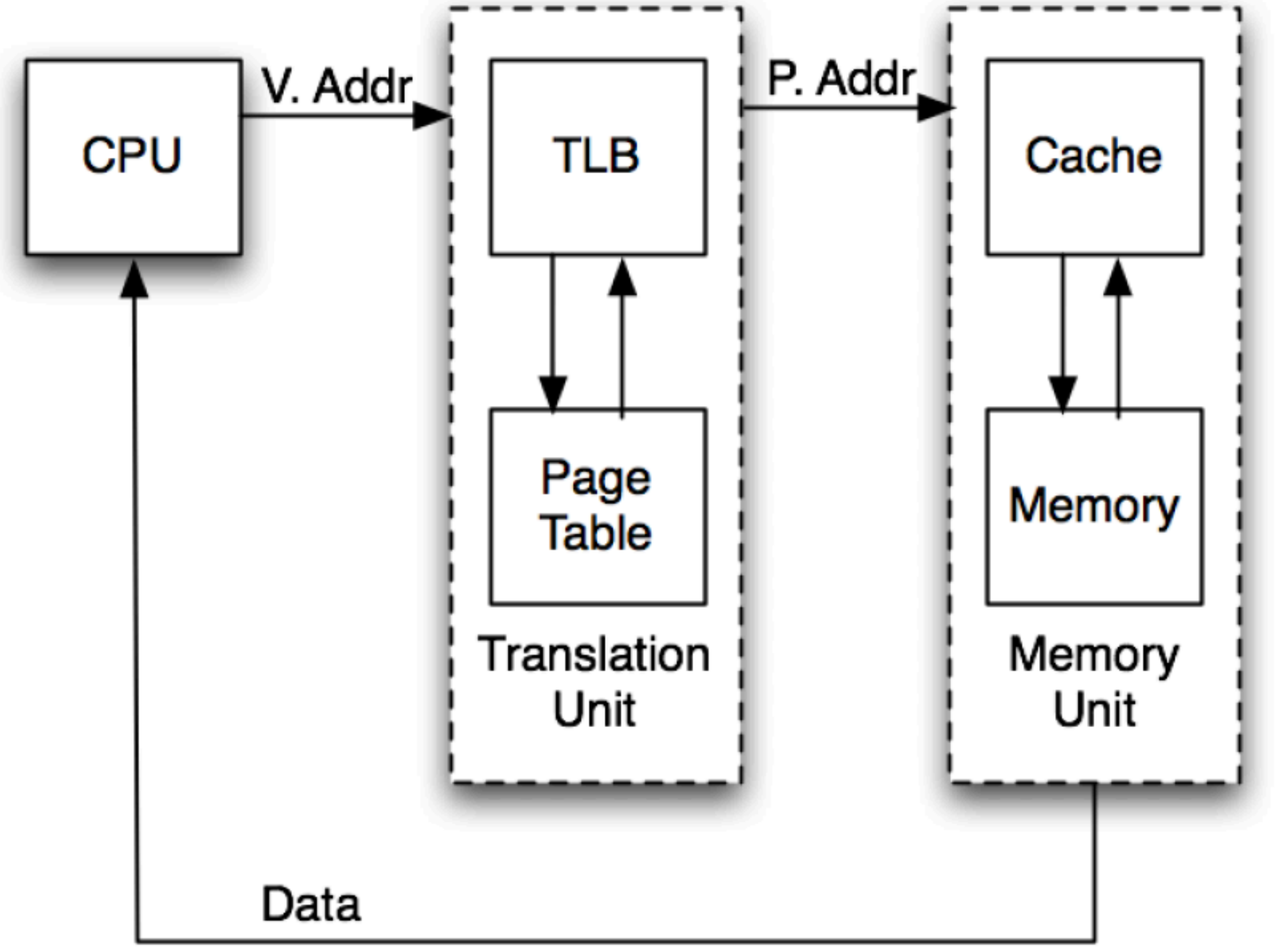
Translation lookaside buffer

TLB

A cache for the page table. Each block is a single page table entry. If an entry is not in the TLB, it's a TLB miss. Assuming fully associative:

<table>
<tr><th rowspan="2">TLB Valid</th><th rowspan="2">Tag (VPN)</th><th colspan="3">Page Table Entry</th></tr>
<tr><th>Page Dirty</th><th>Permission Bits</th><th>PPN</th></tr>
<tr><td colspan="5">— TLB entry —</td></tr>
<tr><td colspan="5">— TLB entry —</td></tr>
</table>

Overall


CPU
V. Addr
TLB
Page Table
Translation Unit
P. Addr
Cache
Memory
Memory Unit
Data

Question

If a page table entry can not be found in the TLB, then a page fault has occurred.

False, the TLB acts as a cache for the page table, so an item can be valid in page table but not stored in TLB. A page fault occurs either when a page cannot be found in the page table or it has an invalid bit.

Question

(Multiple choice) Which of the following things are the Paging capable of while segmenting (base and bound) does not? Circle the letter of the the your choice(s).

A. location independent programming

B. run programs larger than DRAM

C. protection and privacy

D. no (external) memory fragmentation

Solution: B, D

Question

(a) Consider an access pattern to those page tables: 3, 2, 1, 0, 3, 2, 4, 3, 2, 1, 0, 4. How many misses in the TLB will happen if the TLB can hold 3 entries? Which pages are in the TLB in the end? What if theTLB can hold 4 entries? The replacement policy is Least Recently Used (LRU) and the TLB is empty at start.

3 entries: Misses: __________ Entries at end: __________

4 entries: Misses: __________ Entries at end: __________

Solution:
10. 1, 0, 4.
8. 4, 2, 0, 1.

Q&A

THANKS FOR LISTENING