

Memory Mapping

- VM areas initialized by associating them with disk objects.
 - Process is known as *memory mapping*.
- Area can be **backed by file** (i.e., get its initial values from) :
 - *Regular file* on disk (e.g., an executable object file)
 - Initial page bytes come from a section of a file
 - *Anonymous file* (e.g., nothing)
 - First fault will allocate a physical page full of 0's (*demand-zero page*)
 - Once the page is written to (*dirtied*), it is like any other page
- Dirty pages are copied back and forth between memory and a special *swap file*.

Demand paging

- ***Key point:*** no virtual pages are copied into physical memory until they are referenced!

- Known as ***demand paging***

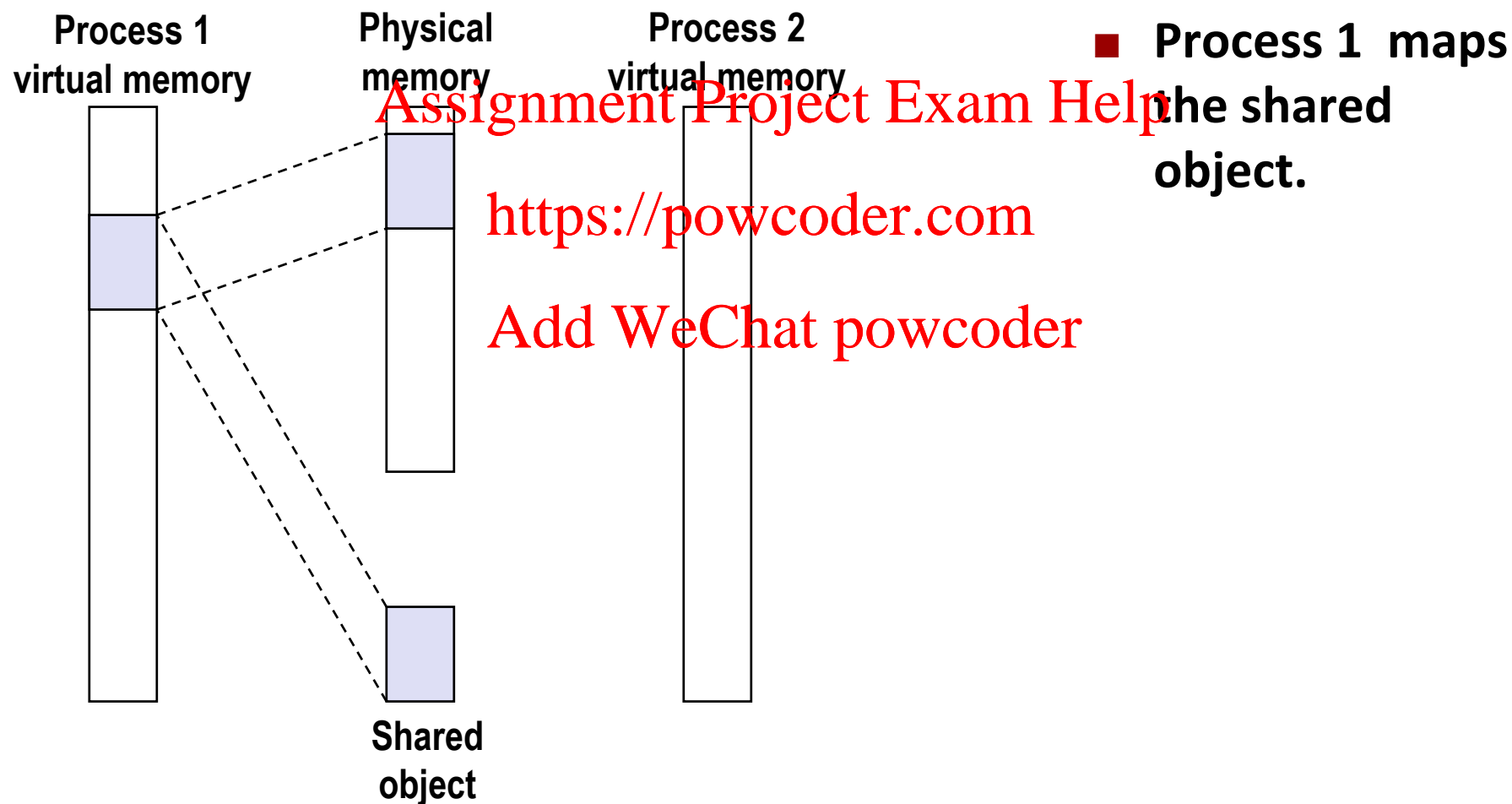
Assignment Project Exam Help

- **Crucial for time and space efficiency**

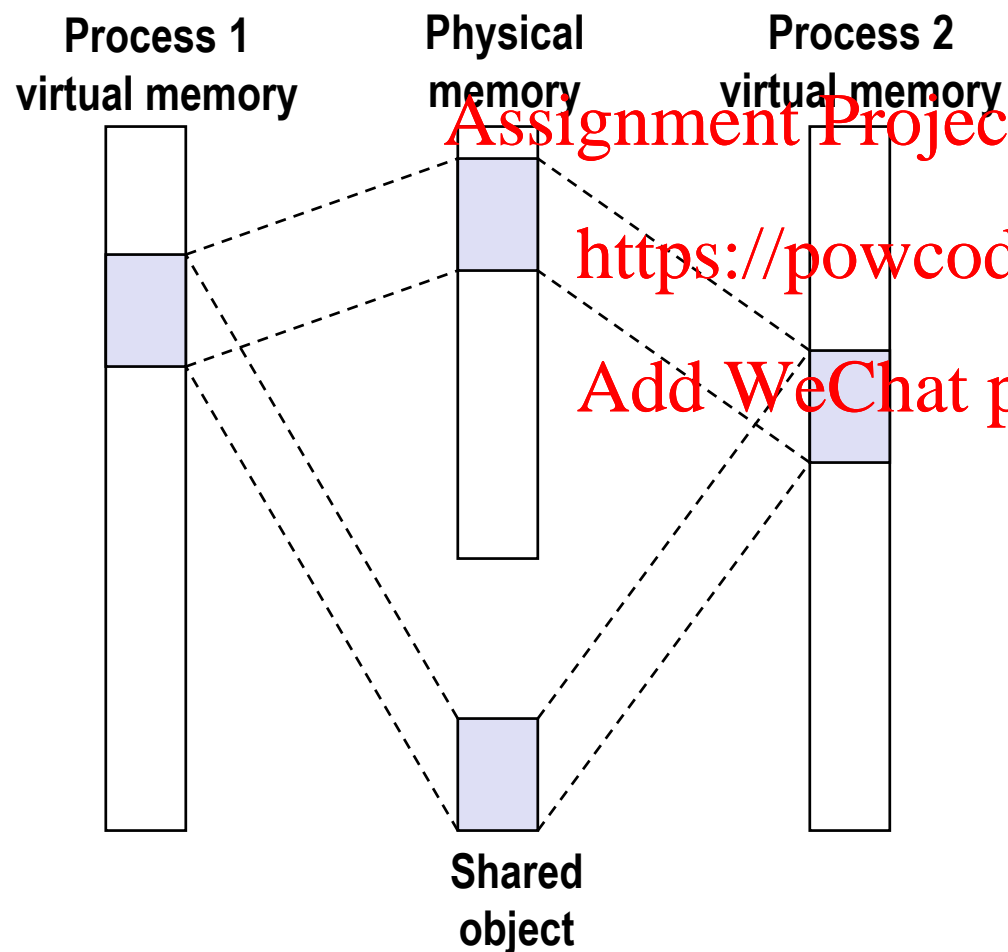
<https://powcoder.com>

Add WeChat powcoder

Sharing Revisited: Shared Objects



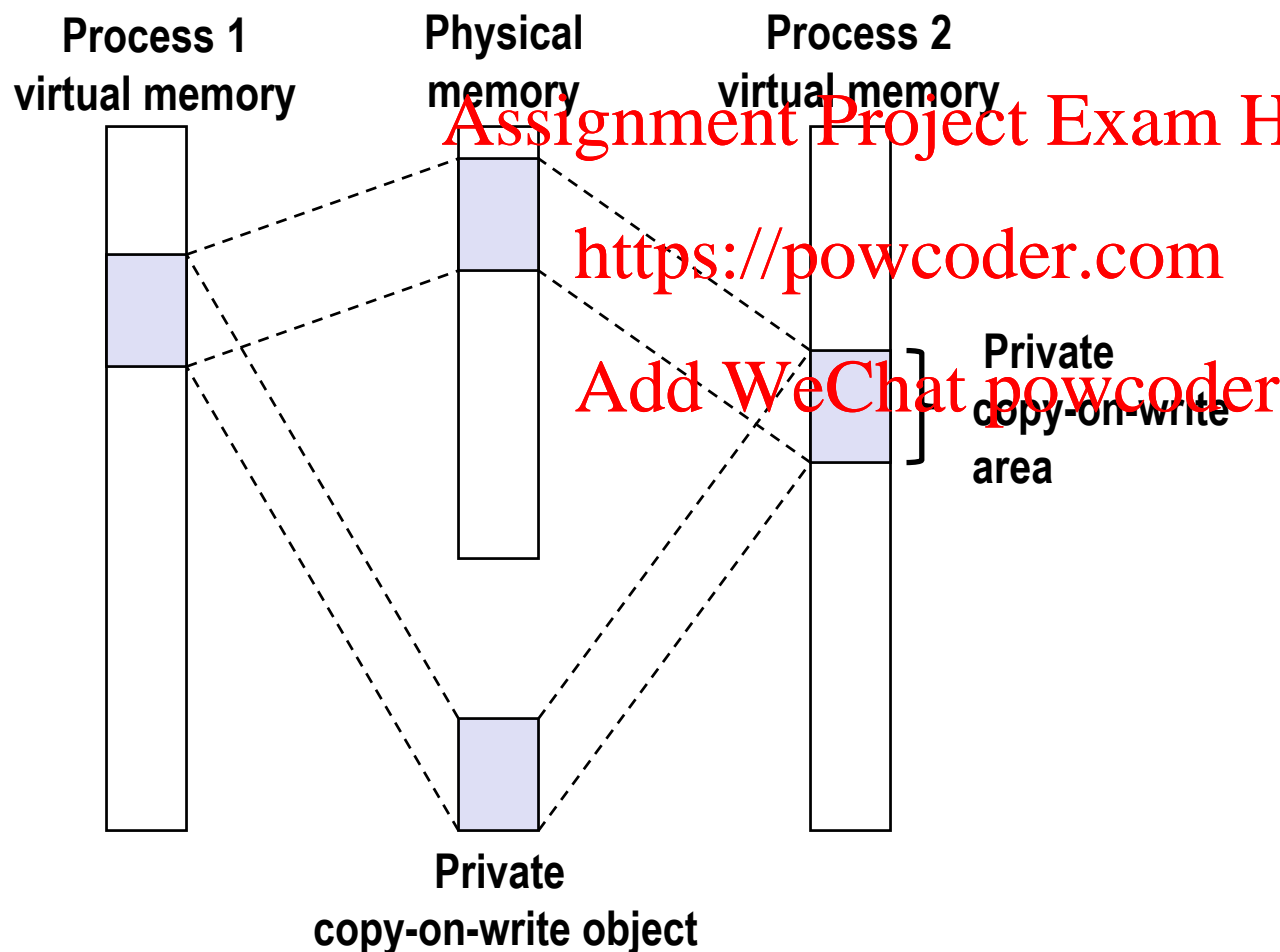
Sharing Revisited: Shared Objects



- Process 2 maps the shared object.
- Notice how the virtual addresses can be different.

Sharing Revisited:

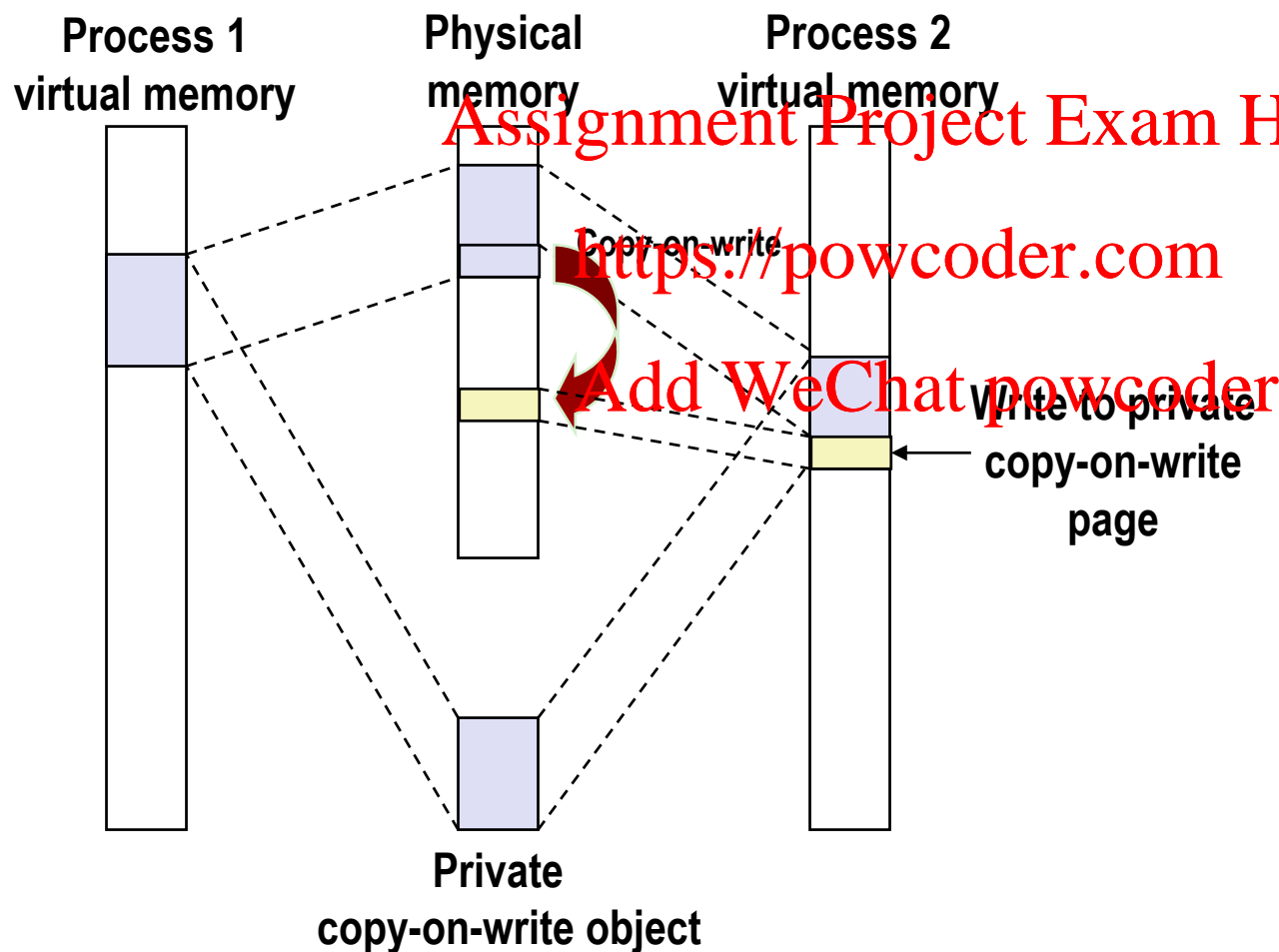
Private Copy-on-write (COW) Objects



- Two processes mapping a **private copy-on-write (COW)** object.
- Area flagged as private copy-on-write
- PTEs in private areas are flagged as read-only

Sharing Revisited:

Private Copy-on-write (COW) Objects



- Instruction writing to private page triggers protection fault.
- Handler creates new R/W page.
- Instruction restarts upon handler return.
- Copying deferred as long as possible!