STEVENS
INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY®

CS 492: Operating Systems

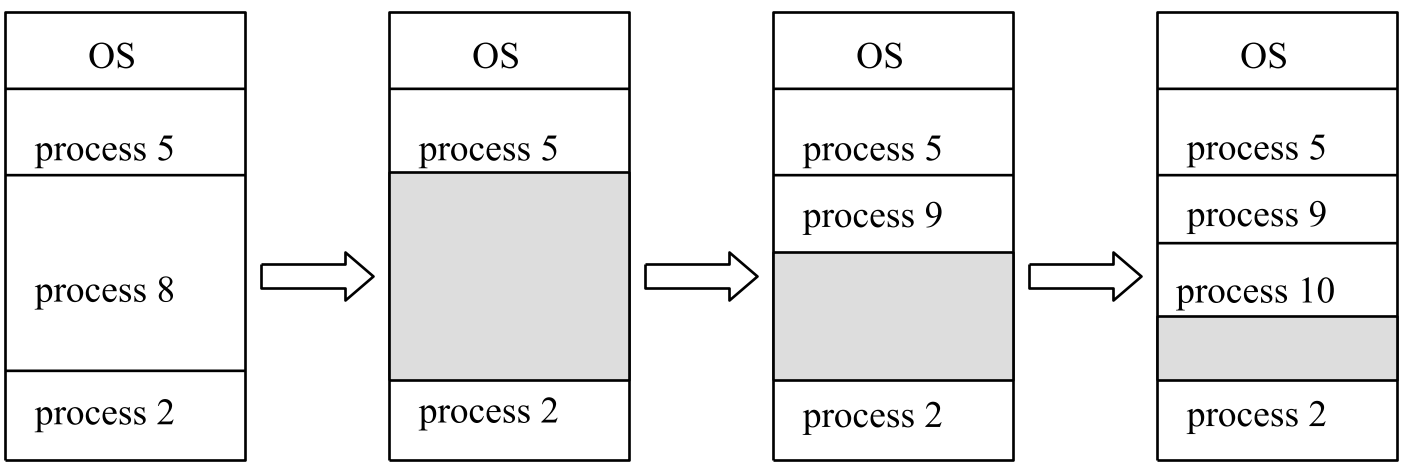
Memory Management

Instructor: Iraklis Tsekourakis

Email: itsekour@stevens.edu

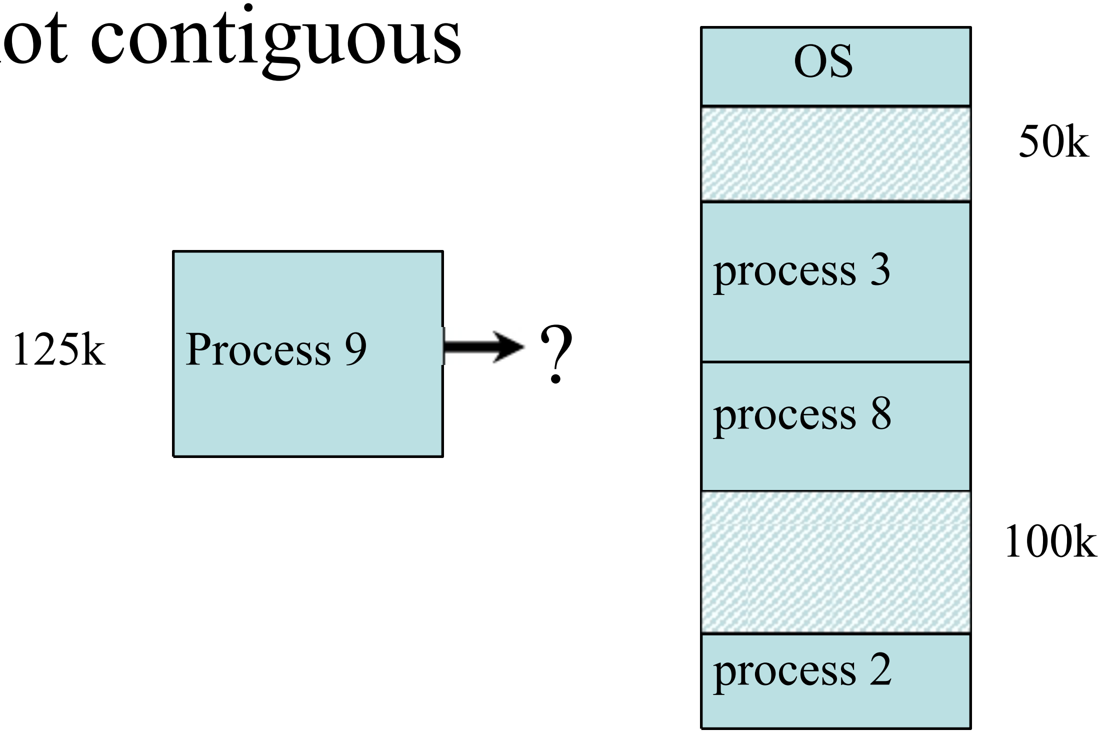
Memory Holes

- Hole – block of available memory; holes of various size are scattered throughout memory
- When a process arrives, it is allocated memory from a hole large enough to accommodate it



External Fragmentation

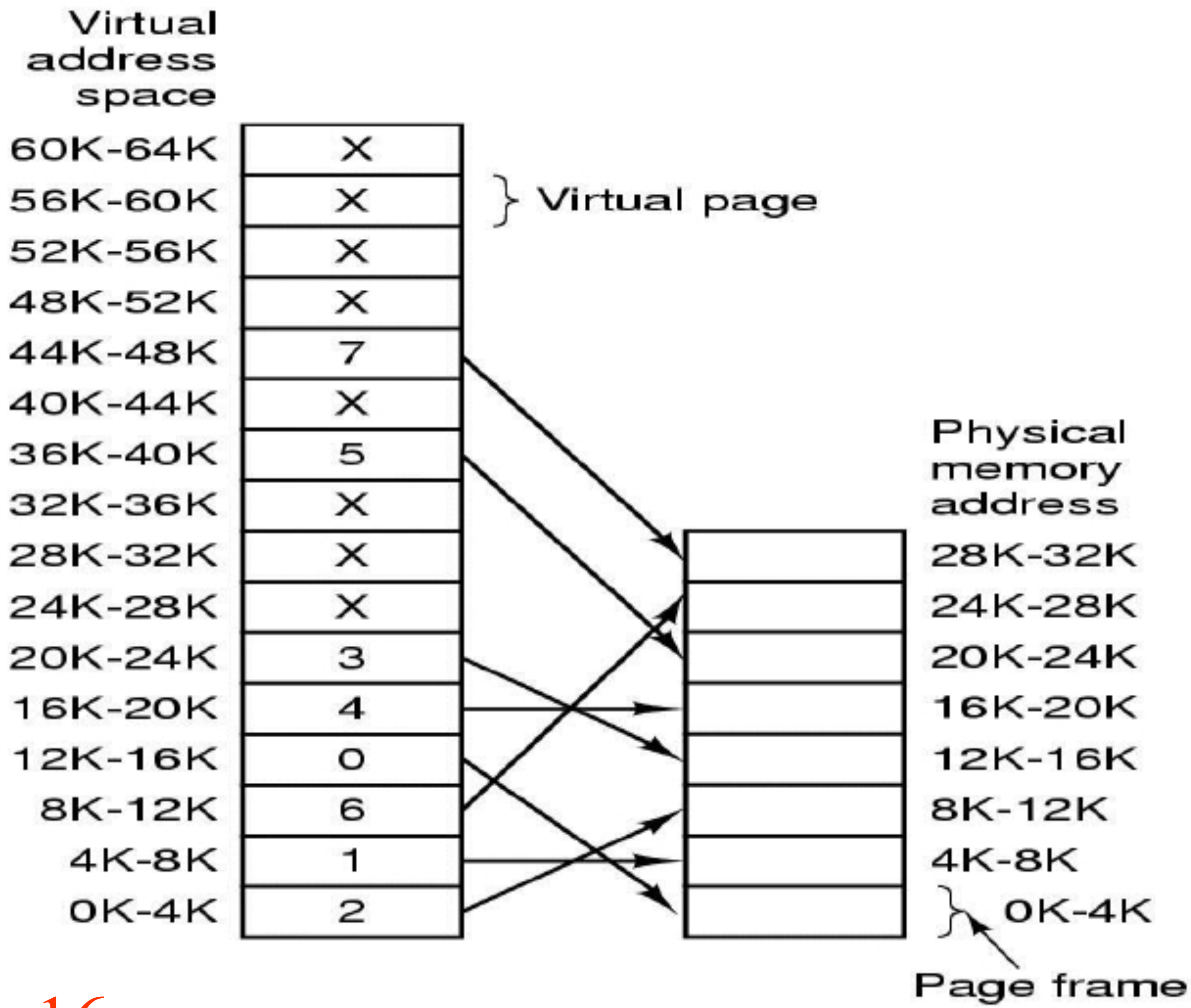
- External fragmentation: total memory space exists to satisfy request but it is not contiguous



Modern technique: Paging

- Solve the external fragmentation problem by using fixed-size chunks of virtual and physical memory
  - Virtual memory unit called a *page*
  - Physical memory unit called a *frame* (or sometimes *page frame*)
  - **Pages and page frames are of the same size**
    - Size is a power of 2
    - Typically 4-64KB
- Virtual address space: a sequence of ***pages***
- Physical memory: a sequence of ***page frames***

Virtual Addresses V.S. Physical Memory



16 pages

8 page frames

Question

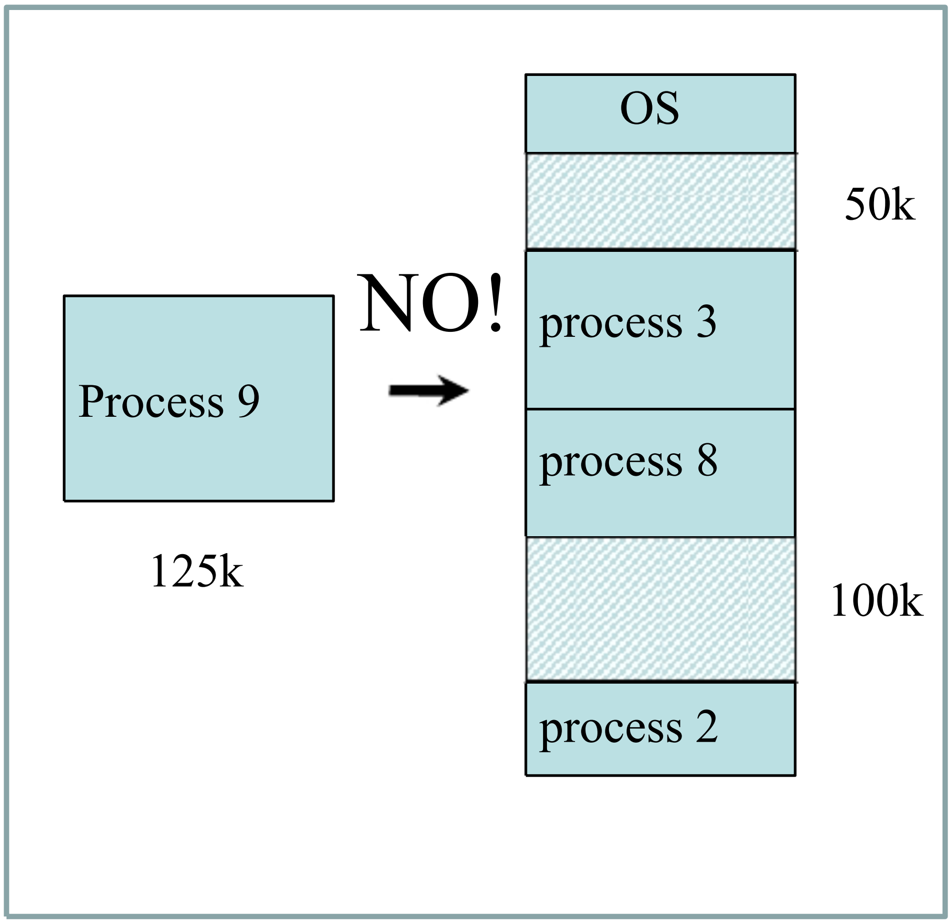
Using the previous page table, give the physical address corresponding to each of the following virtual addresses:

(a) 20

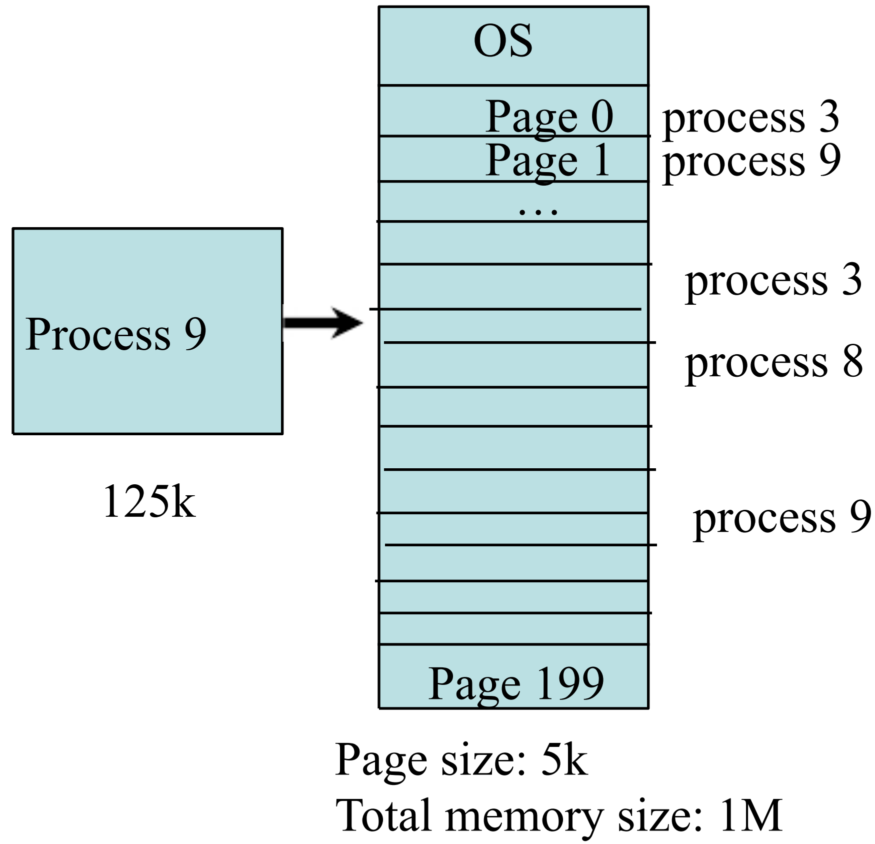
(b) 4100

(c) 8300

How Do Pages Fix External Fragmentation Problem?



External Fragmentation

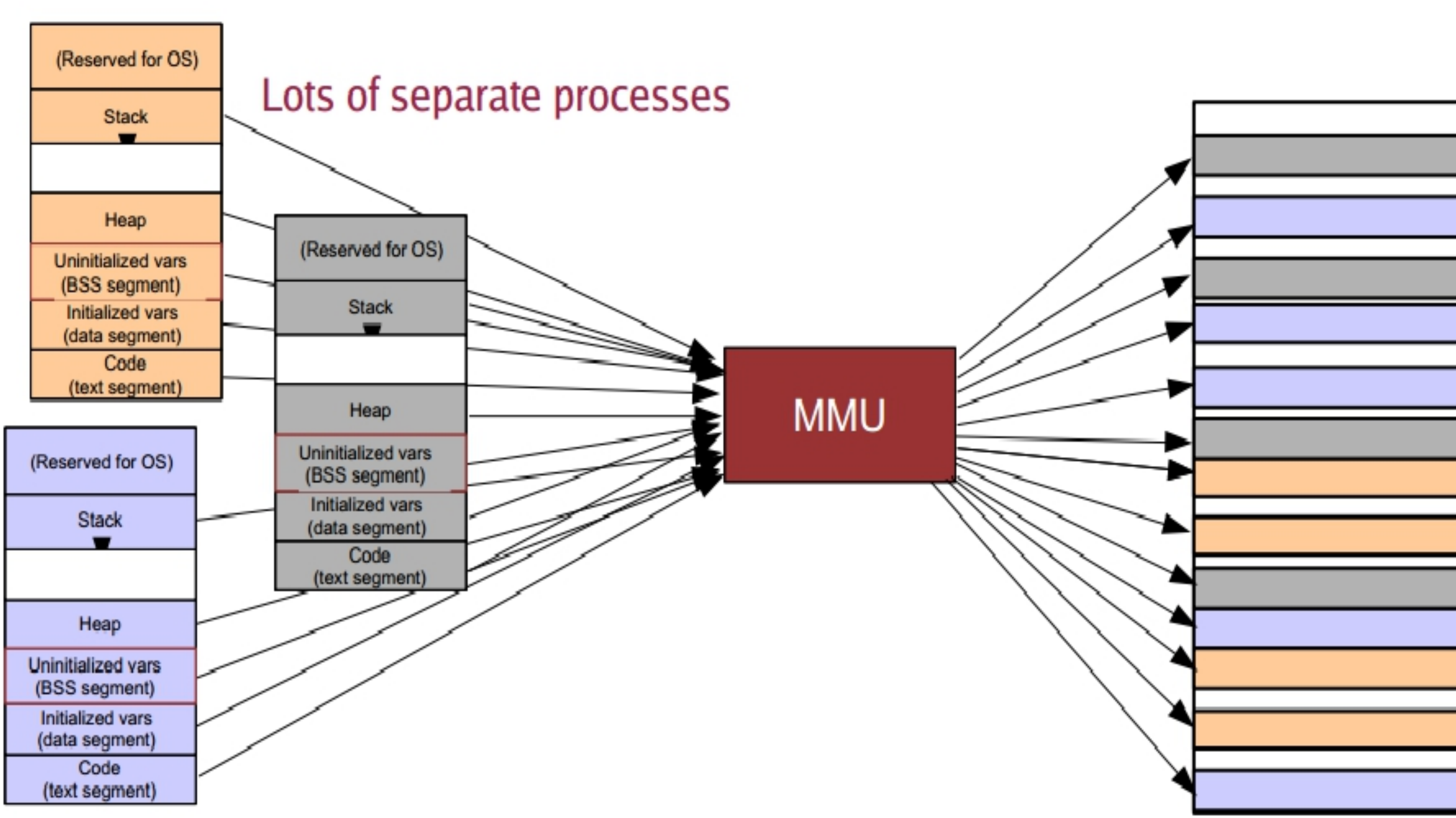


Pages of Virtual Memory

Application Perspective

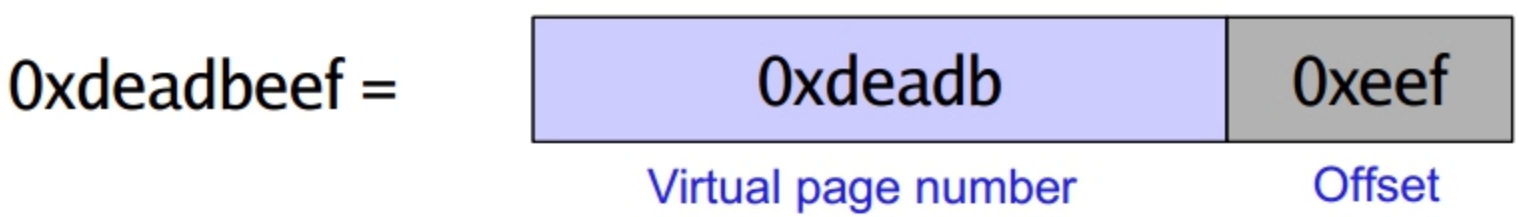
- Application believes it has a single, contiguous address space ranging from 0 to $2^P - 1$ bytes
  - Where P is the number of bits in a pointer (e.g., 32 bits)
- In reality, virtual pages are scattered across physical memory
  - This mapping is invisible to the program, and not even under it's control!

Illustration of Application Perspective



Address Translation

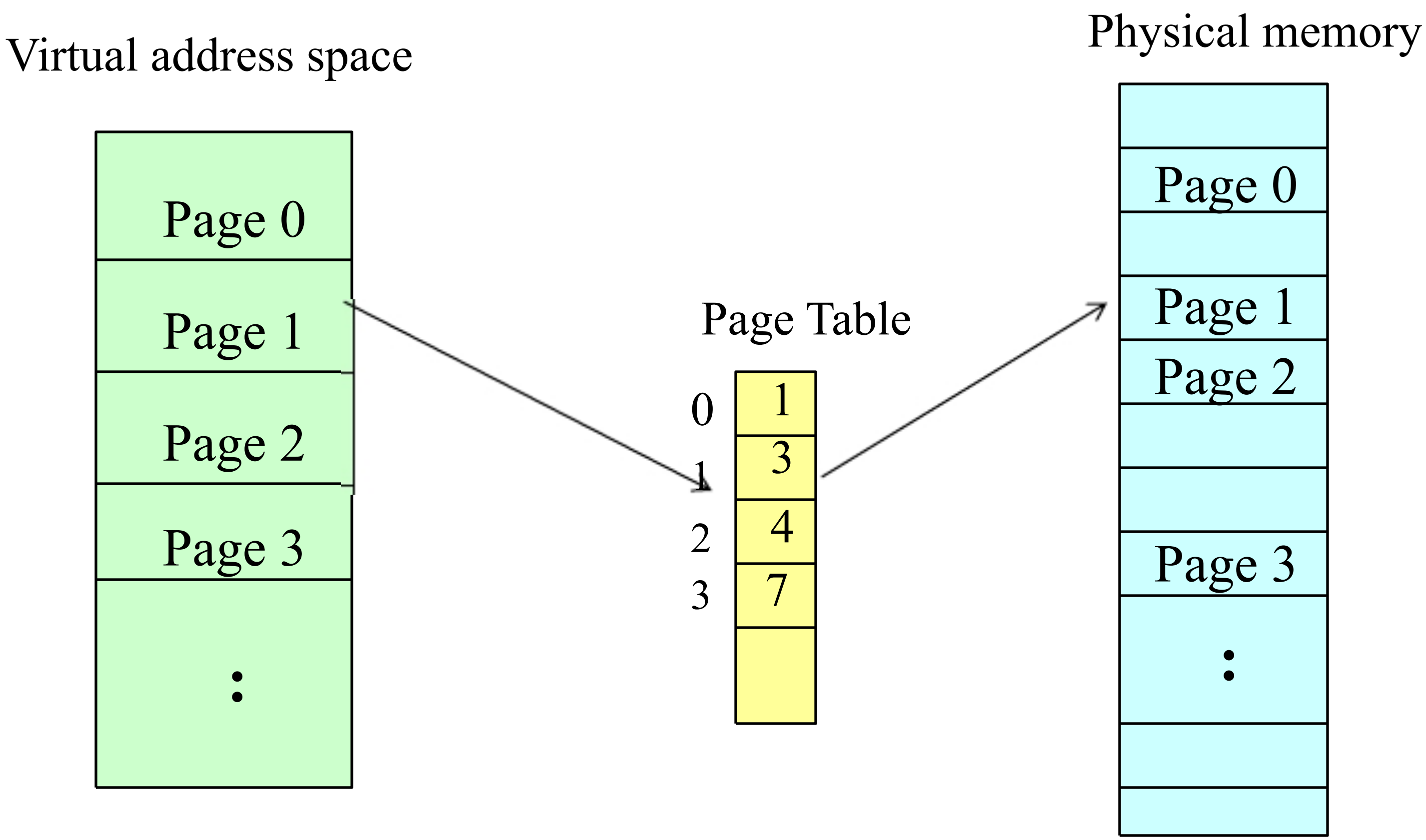
- Any page can be translated to any page frame
- Translation is done by MMU
  - Virtual address is broken into: (1) a *virtual page number*, and (2) an *offset*
  - Mapping from virtual page to physical frame provided by a *page table*



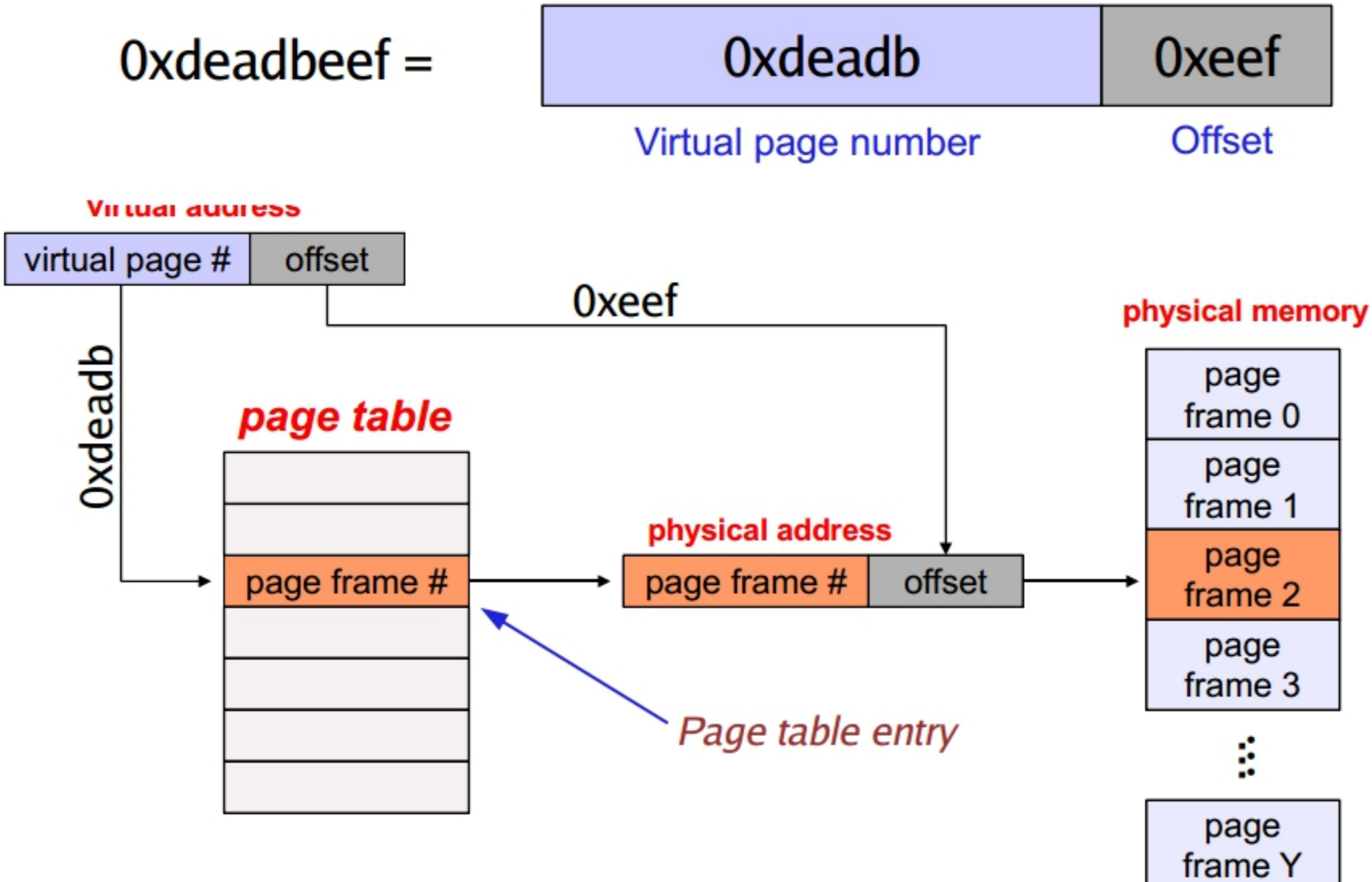
Page Table

- MMU translates virtual address to physical address via the page table
- Number of entries in page table = number of virtual pages
- One page table per process
  - Located in memory (either kernel's physical memory or process' virtual memory)

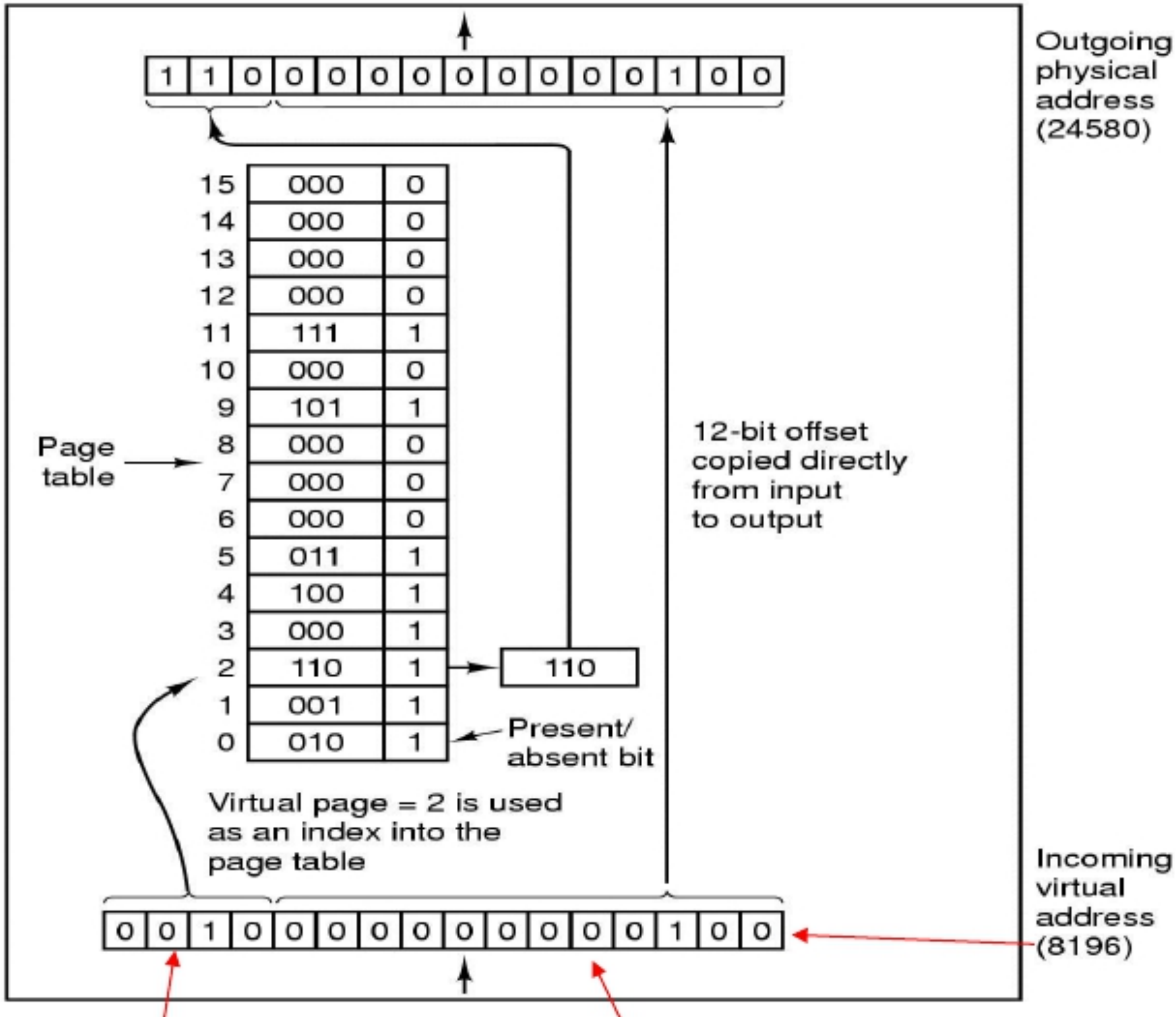
Example of Page Table



Details of Address Translation by Page Table



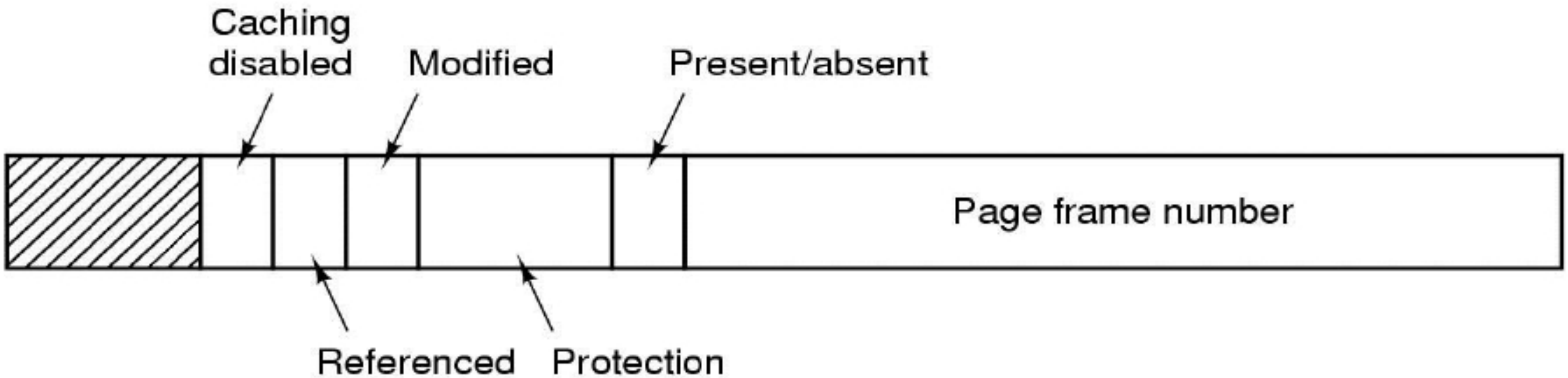
Example: Address Translation



16 virtual pages: |p|=4

Page size 4 KB: |d|=12

Page Table Entry (PTE)



- Page Frame # - physical page frame where virtual page is loaded
- Present/Absent bit – Is page loaded in main memory?
- Protection – read, write, executable
- “dirty bit” – has page been modified?
  - If yes, then need to write back to disk when page is swapped out of physical memory
- Referenced – set when page is accessed
  - Helps OS decide whether or not to swap page out
- Disable caching – if set, do not use cached copy
  - Cached copy may be invalid if page is mapped to I/O device and will change often (memory-mapped I/O)