IOWA STATE UNIVERSITY
Department of Electrical and Computer Engineering

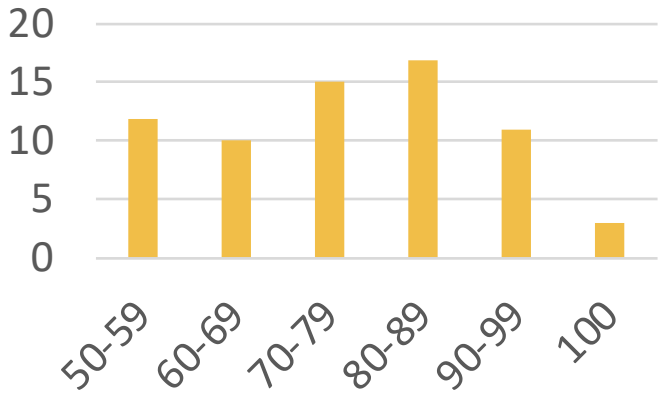
Lecture 21: Page Table

Agenda

- Recap
- Page Table
  - Location of Page Table
  - Page Table Entry (PTE)

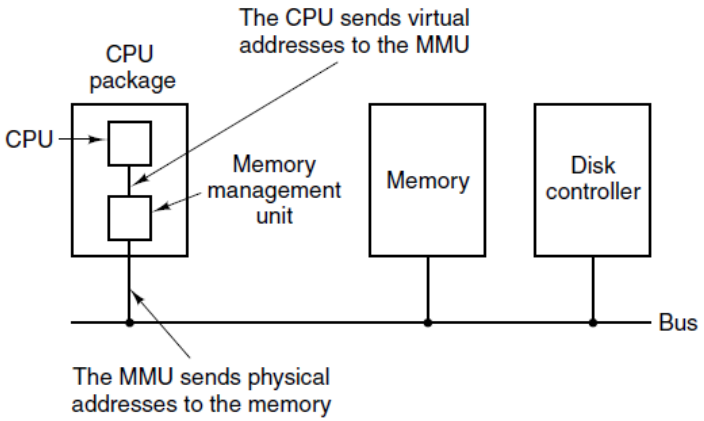
Recap

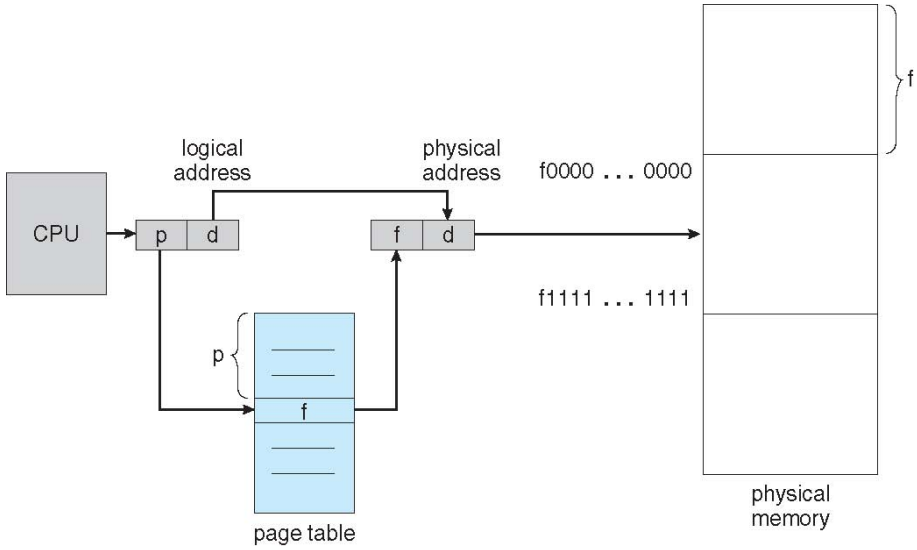
- Midterm 1 feedback
  - Observations
    - Top scorers
      - attended the class regularly
      - participated in Q&A actively
    - Those sent emails to me about class participation are above average in general
      - followed the instructions
  - Don’t lose heart if you did not do well
    - score on paper does not necessarily reflect your potential
    - You may want to adjust your study method
      - Attend (more) class, ask (more) questions, ...
      - Welcome to visit my office hours



Recap

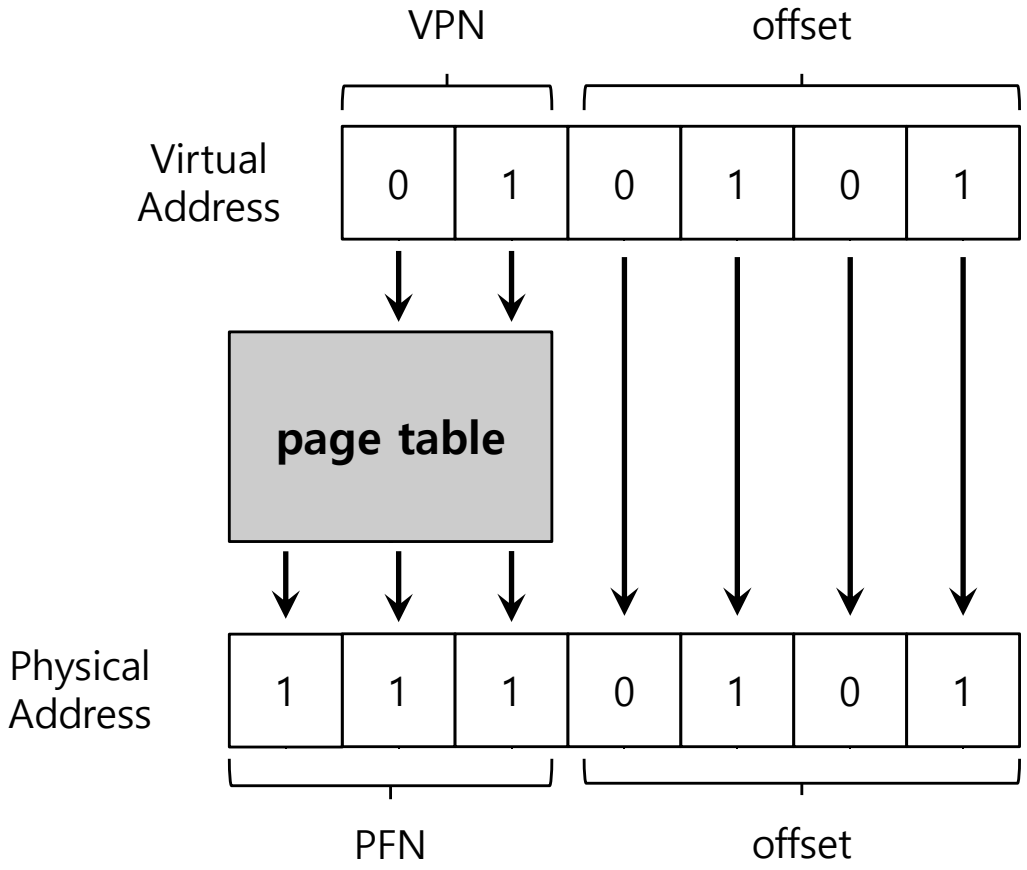
- Address Translation via **Page Table**
  - Each virtual address is divided into two parts:
    - VPN: virtual page number (p)
      - used as an index into the **page table**
    - Offset: offset within the page (d)





Recap

- Address Translation via **Page Table**
  - Example: the virtual address 21 in 64-byte address space
    - 64 = 2^6
    - 21: 0x010101



Agenda

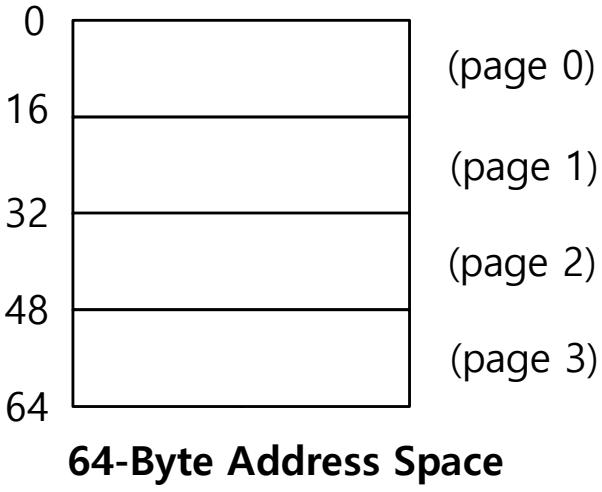
- ~~Recap~~
- Page Table
  - Location of Page Table
  - Page Table Entry (PTE)

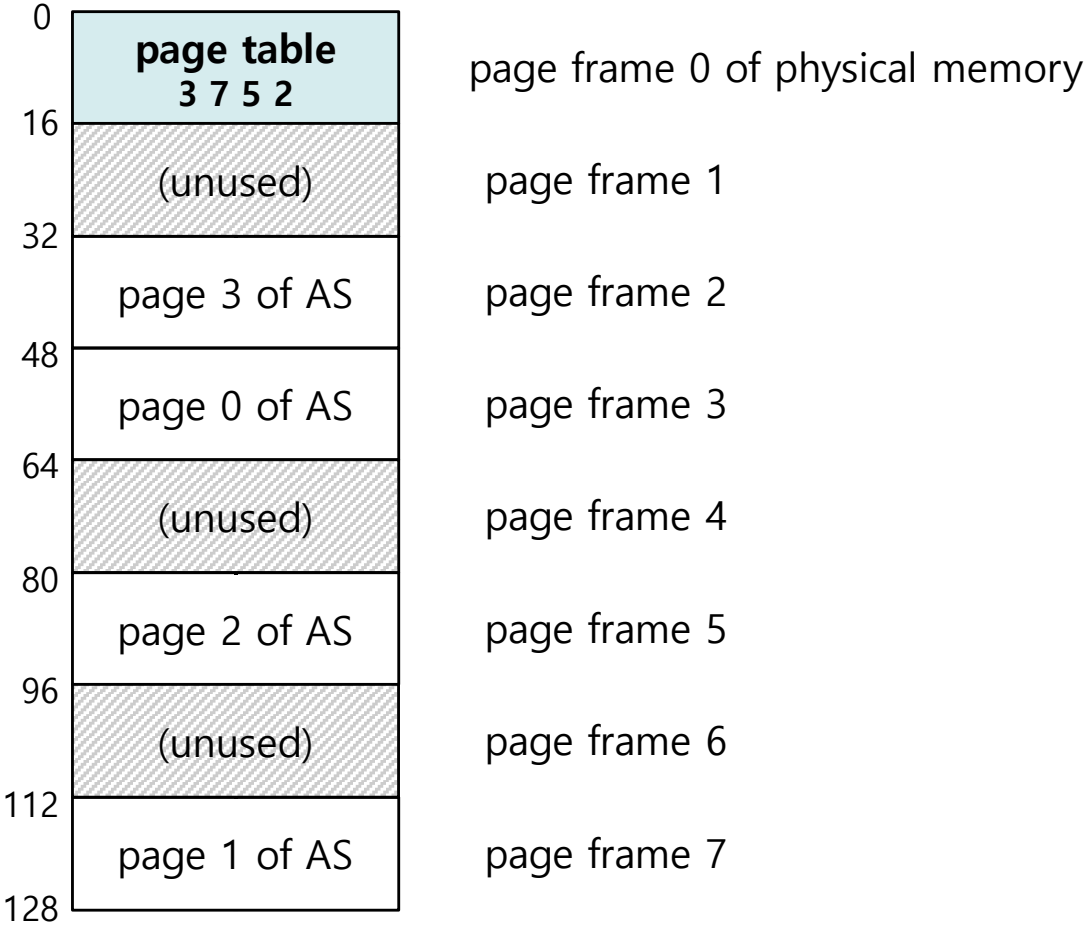
Where Are Page Tables Stored?

- The page table for each process is stored in physical memory
  - accessible by kernel
  - Page-table base register (PTBR) points to the page table
  - Page-table length register (PTLR) indicates size of the page table

Where Are Page Tables Stored?

- The page table for each process is stored in physical memory





What Is in the Page Table?

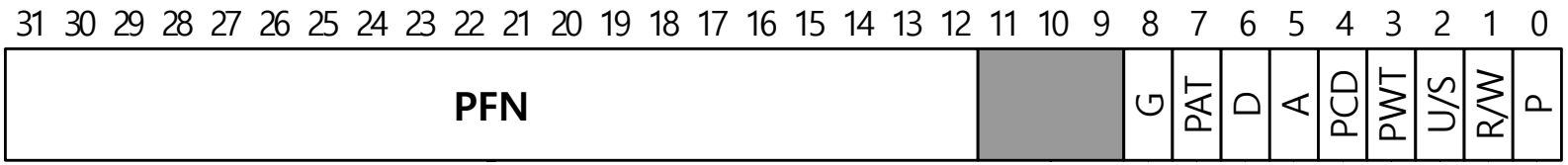
- The page table is just a **data structure** that is used to map the virtual address to physical address.
  - Simplest form: a linear page table, an array
- The OS **indexes** the array by VPN, and looks up the page-table entry
  - Each entry must contain the base address of each page in the physical memory
  - may also contain additional bits for other purposes (e.g., protection)

What Is in the Page Table?

- Common Flags Of Page Table Entry (PTE)
  - **Valid Bit**: Indicating whether the particular translation is valid
  - **Protection Bit**: Indicating whether the page could be read from, written to, or executed from
  - **Present Bit**: Indicating whether this page is in physical memory or on disk(swapped out)
  - **Dirty Bit**: Indicating whether the page has been modified since it was brought into memory
  - **Reference Bit(Accessed Bit):** Indicating that a page has been accessed

What Is in the Page Table?

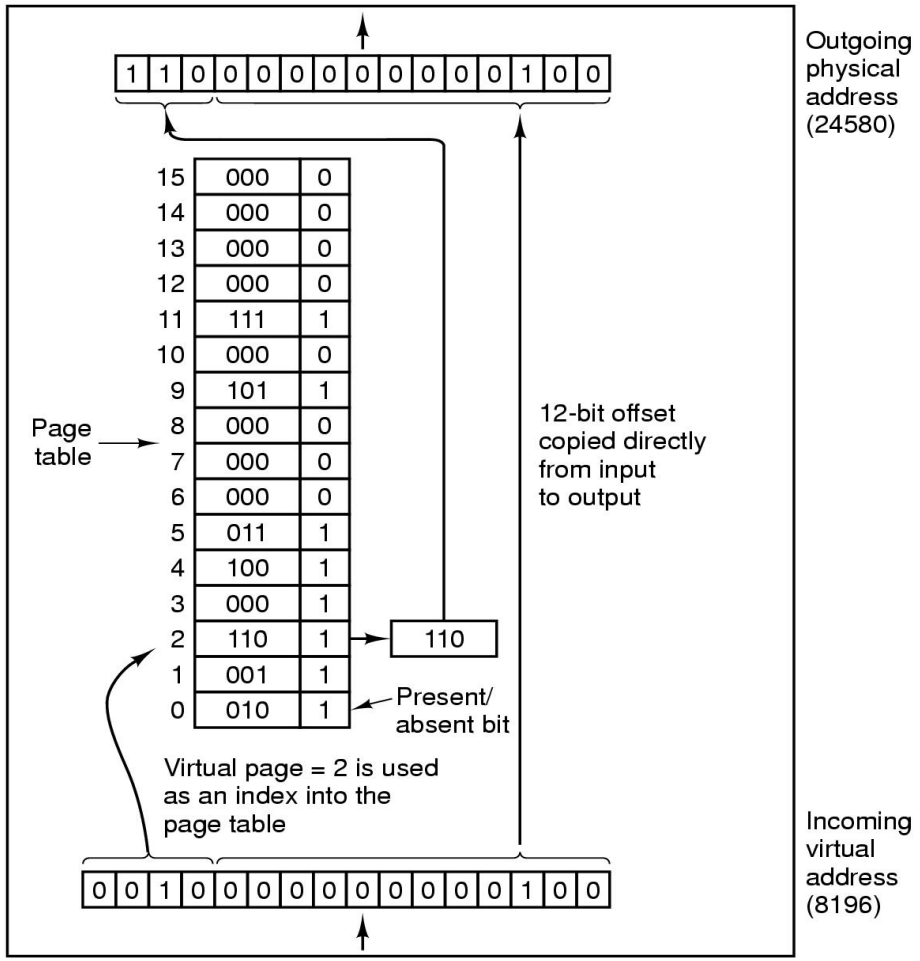
- Example: x86 Page Table Entry (PTE)



- P: present
- R/W: read/write bit
- U/S: supervisor
- A: accessed bit
- D: dirty bit
- PFN: the page frame number

What Is in the Page Table?

- Address translation using a page table

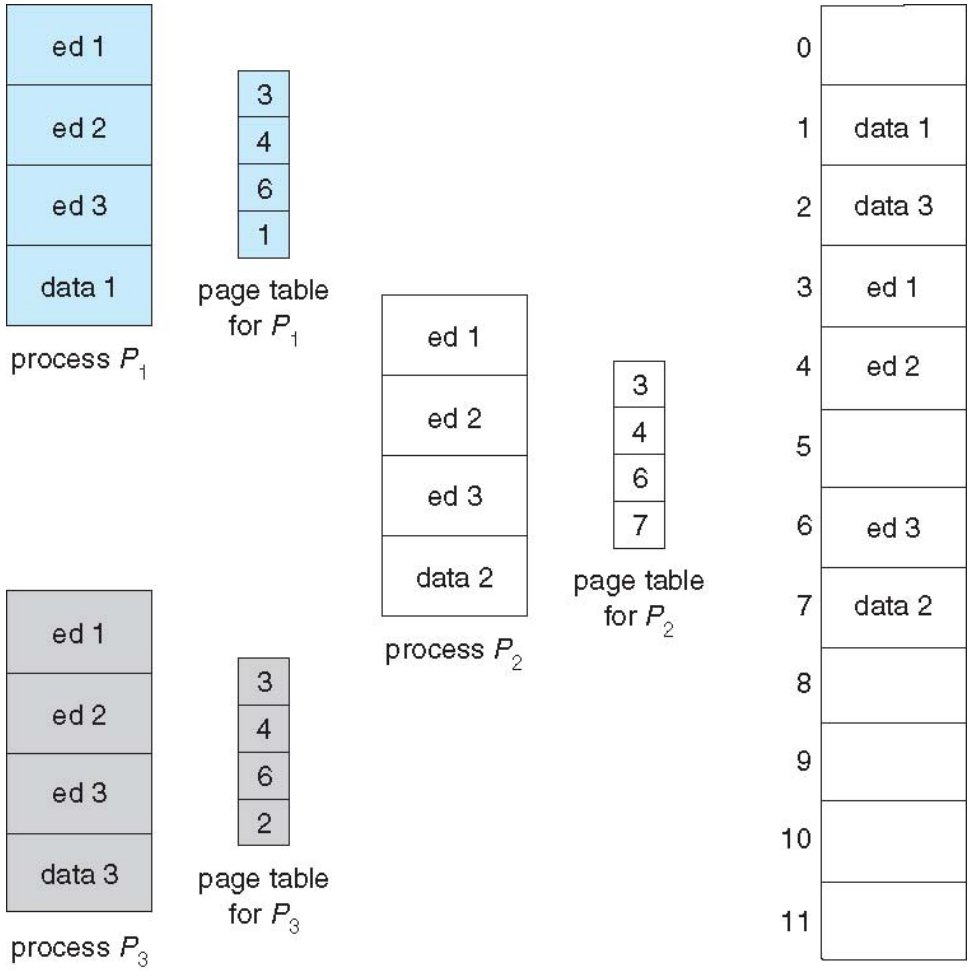


Shared Pages

- One page in the physical memory is mapped to the address spaces of multiple processes
  - Often used for code sharing
    - One copy of code shared among processes (i.e., text editors, compilers, window systems)
    - Similar to multiple threads sharing the same process space
  - Also useful for inter-process communication if sharing of read-write pages is allowed
  - Each process keeps a separate copy of the private code/data

Shared Pages

- Example of shared pages



What if requested page not in memory?

- **Page Fault**
  - MMU reports a page fault to CPU
  - CPU gives control to the OS (page fault handler routine)
  - OS fetches a page from the disk
    - May needs to evict an existing page from memory (page replacement policy)
  - Instruction is restarted

Potential Issues of Paging

- Time
  - For every user memory access, one additional memory access is needed (for accessing page table in memory)
- Space
  - Page tables can get awfully large
    - 32-bit address space (4GB) with 4KB pages
      - 12 bits for offset within a page (4K=2^12)
      - 20 bits for VPN (32 – 12 = 20)
      - $4MB = 2^{20}\ entries * 4\ Bytes\ per\ page\ table\ entry$
    - Each process needs to have a page table!

Agenda

- ~~Recap~~
- ~~Page Table~~
  - ~~Location of Page Table~~
  - ~~Page Table Entry (PTE)~~

Questions?

*acknowledgement: slides include content from “Modern Operating Systems” by A. Tanenbaum, “Operating Systems Concepts” by A. Silberschatz etc., “Operating Systems: Three Easy Pieces” by R. Arpaci-Dusseau etc., and anonymous pictures from internet.