Lab 9: Paging

Sejong Yoon, Ph.D.

References:

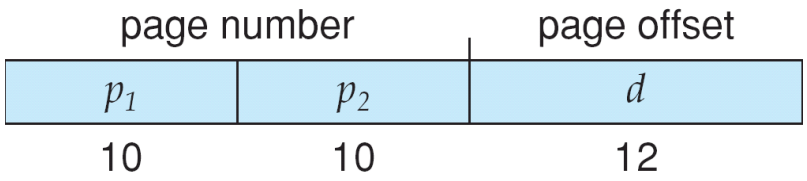
- Silberschatz, et al. *Operating System Concepts* (9e), 2013
- Materials from OS courses offered at TCNJ (Dr. Jikai Li), Princeton, Rutgers, Columbia (Dr. Junfeng Yang), Stanford, MIT, UWisc, VT

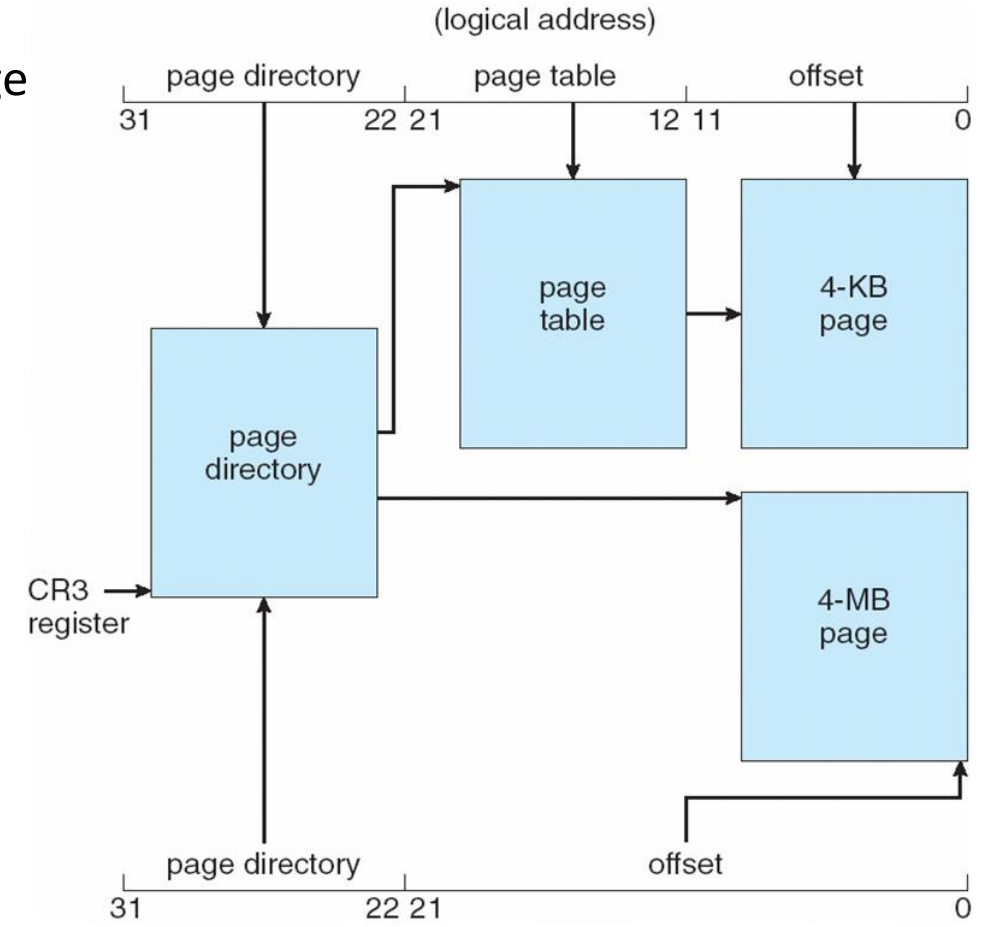
Agenda

- Review: x86 paging
- Exercise

Review: IA-32 Paging Architecture

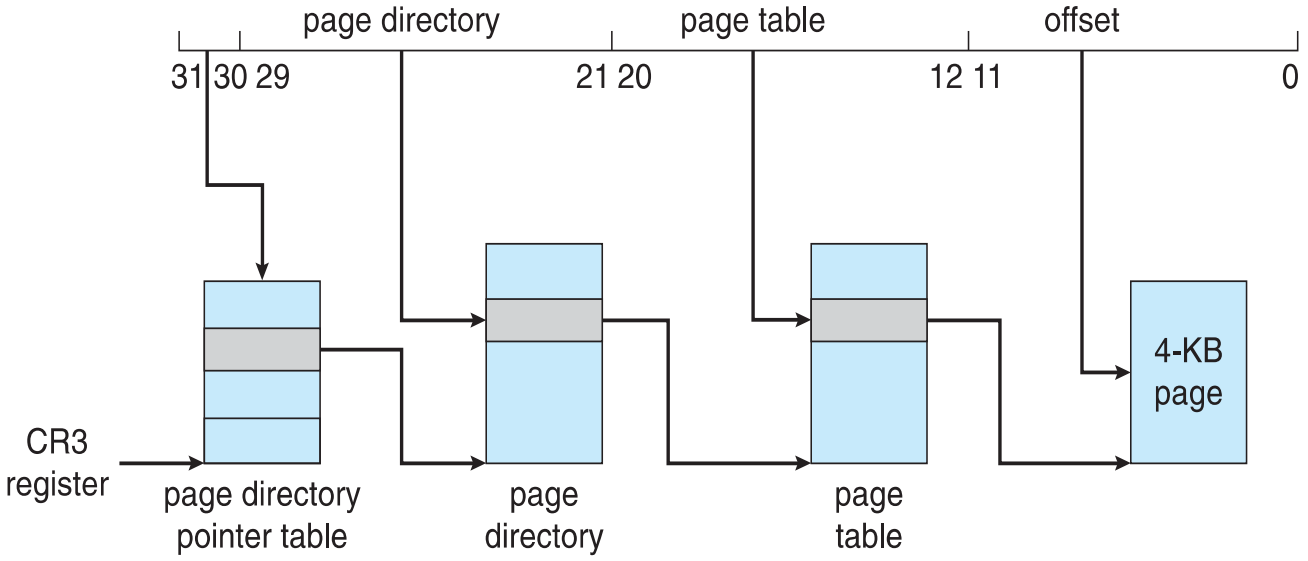
- Allows either 4 KB or 4 MB page





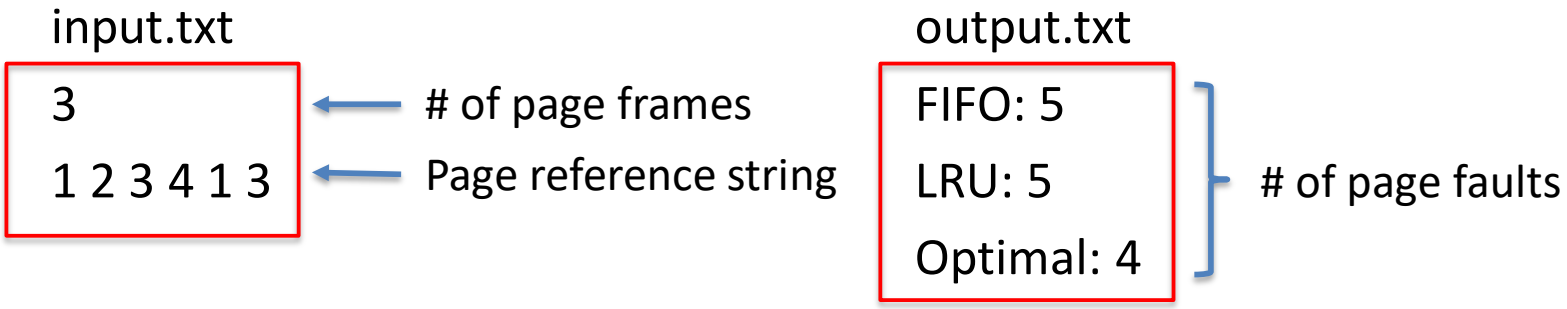
Review: IA-32 Page Address Extensions

- 32-bit address limits led Intel to create **page address extension** (**PAE**), allowing 32-bit apps access to more than 4GB of memory space
  - Paging went to a 3-level scheme
  - Top two bits refer to a **page directory pointer table**
  - Page-directory and page-table entries moved to 64-bits in size
  - Net effect is increasing address space to 36 bits – 64GB of physical memory



Exercise 9.39

- Write a program that implements the FIFO, LRU, and optimal page replacement algorithms presented in this chapter. First, generate a random page-reference string where page numbers range from 0 to 9. Apply the random page-reference string to each algorithm, and record the number of page faults incurred by each algorithm. Implement the replacement algorithms so that the number of page frames can vary from 1 to 7. Assume that demand paging is used. Your program should read input.txt from the same directory and generate output.txt in the same directory. Formats are like below:



Lab 9 assignment

- Add a comment to the beginning of your source code containing your name, the name of the course, and the title of the assignment:

  /** John Smith

  CSC345-01

  Lab 9 Exercise 1 */
- Rename your source file into lab09_ex1.c
- Prepare Makefile that compiles your source codes into object code lab09_ex1
- Read (again) chapters 1 and 2 from xv6_book.pdf
  - Write one-page report (11 pt, single spaced, US letter size) describing your understanding on address space and paging of xv6 operating system, and discuss its relationship to concepts learned in class
- Zip your source file, Makefile, **and report pdf** into lab09.zip
- Submit your zip file via Canvas