University of Pittsburgh

Introduction to Operating Systems
CS 1550

Spring 2023

Sherif Khattab

ksm73@pitt.edu

(Some slides are from **Silberschatz, Galvin and Gagne ©2013)**

Announcements

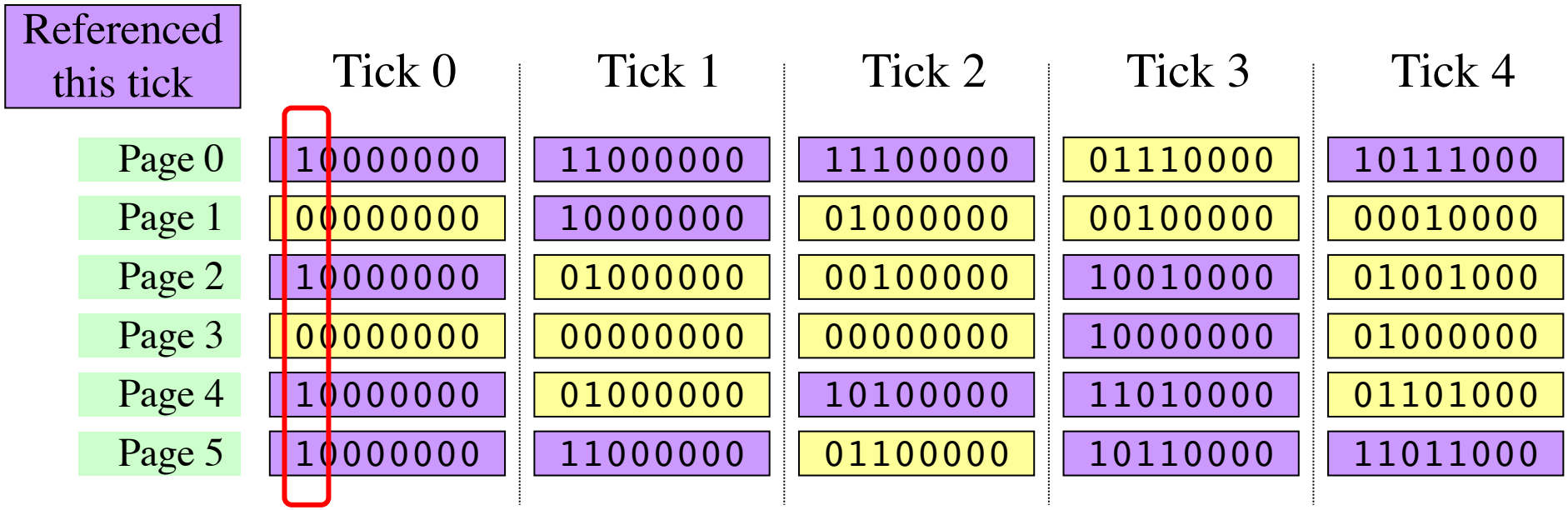
- Upcoming deadlines
  - Quiz 2 and Homework 8 due this Friday
    - 2 extra attempts for HW 8
  - Lab 3 is due on Tuesday 3/28 at 11:59 pm
  - Project 3 is due Friday 4/7 at 11:59 pm

Previous Lecture …

- Page replacement algorithms
  - OPT, NRU, FIFO, Second Chance, CLOCK, LRU, NFU

Aging replacement algorithm

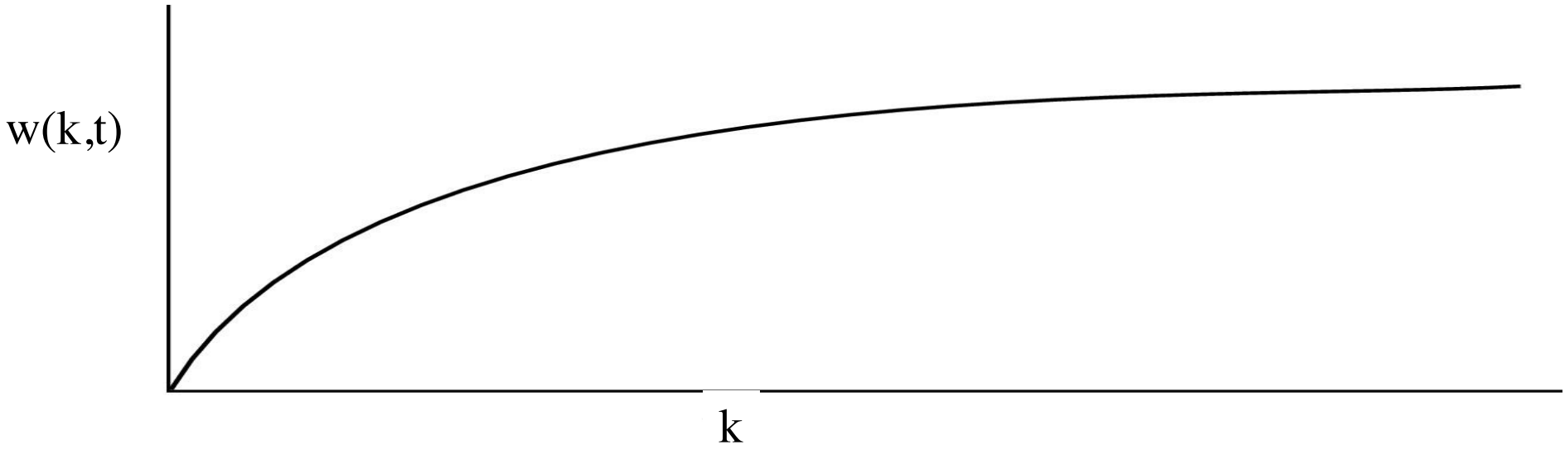
- Reduce counter values over time
  - Divide by two every clock cycle (use right shift)
  - More weight given to more recent references!
- Select page to be evicted by finding the lowest counter value
- Algorithm is:
  - Every clock tick, shift all counters right by 1 bit
  - On reference, set leftmost bit of a counter (can be done by copying the reference bit to the counter at the clock tick)



Working set

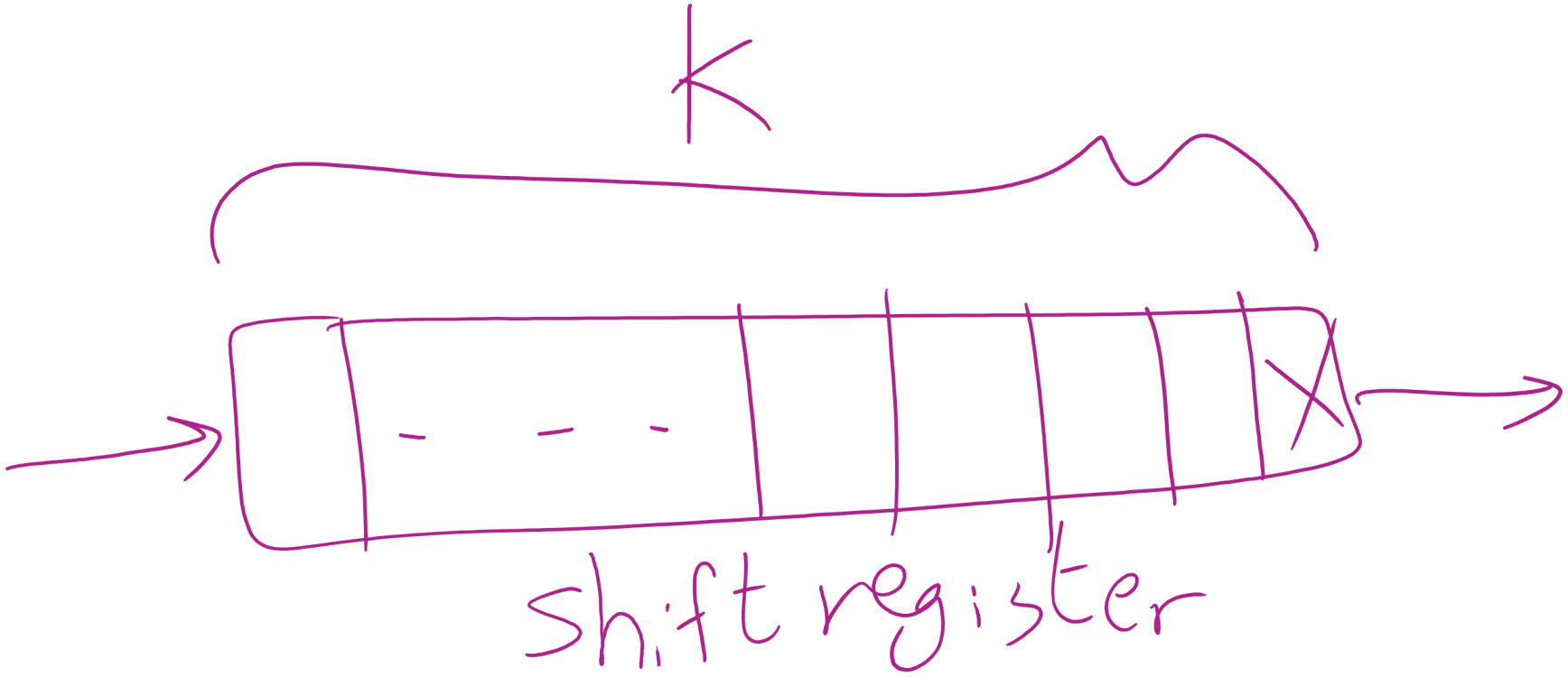
- *Demand paging*: bring a page into memory when it's requested by the process
- How many pages are needed?
  - Could be all of them, but not likely
  - Instead, processes reference a small set of pages at any given time—*locality of reference*
  - Set of pages can be different for different processes or even different times in the running of a single process
- Set of pages used by a process in a given interval of time is called the *working set*
  - If entire working set is in memory, no page faults!
  - If insufficient space for working set, thrashing may occur
  - Goal: keep most of working set in memory to minimize the number of page faults suffered by a process

How big is the working set?

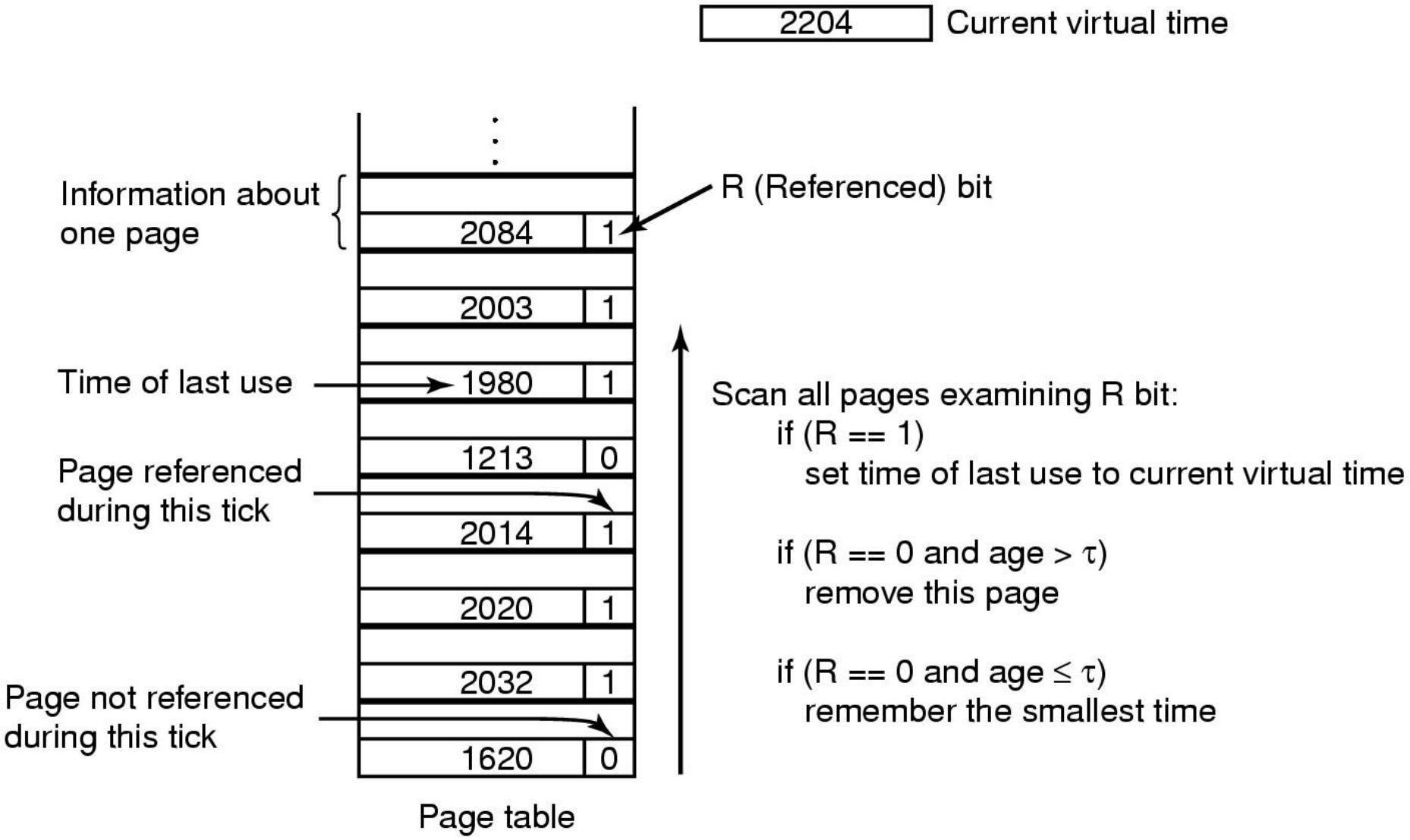


- Working set is the set of pages used by the *k* most recent memory references
- w(k,t) is the size of the working set at time *t*
- Working set may change over time
  - Size of working set can change over time as well…

Keeping track of the Working Set



Working set page replacement algorithm



Summary

- Page replacement algorithms

Algorithm	Comment
OPT (Optimal)	Not implementable, but useful as a benchmark
NRU (Not Recently Used)	Crude
FIFO (First-In, First Out)	Might throw out useful pages
Second chance	Big improvement over FIFO
Clock	Better implementation of second chance
LRU (Least Recently Used)	Excellent, but hard to implement exactly
NFU (Not Frequently Used)	Poor approximation to LRU
Aging	Good approximation to LRU, efficient to implement
Working Set	Somewhat expensive to implement
WSClock	Implementable version of Working Set

Algorithm Simulation

- How to simulate page replacement algorithms
  - FIFO/Clock
  - LRU, OPT

How is modeling done?

- Generate a list of references
  - Artificial (made up)
  - Trace a real workload (set of processes)
- Use an array (or other structure) to track the pages in physical memory at any given time
  - May keep other information per page to help simulate the algorithm (modification time, time when paged in, etc.)
- Run through references, applying the replacement algorithm
- Example: FIFO replacement on reference string 0 1 2 3 0 1 4 0 1 2 3 4
  - Page replacements highlighted in yellow

Page referenced	0	1	2	3	0	1	4	0	1	2	3	4
Youngest page	0	1	2	3	0	1	4	4	4	2	3	3
		0	1	2	3	0	1	1	1	4	2	2
Oldest page			0	1	2	3	0	0	0	1	4	4

Interactive Simulation Tool

- https://sim-50.github.io/cs-tools/

FIFO with 3 frames