# LECTURE 2.6 FRAME ALLOCATION AND EXAMPLES

COP4600

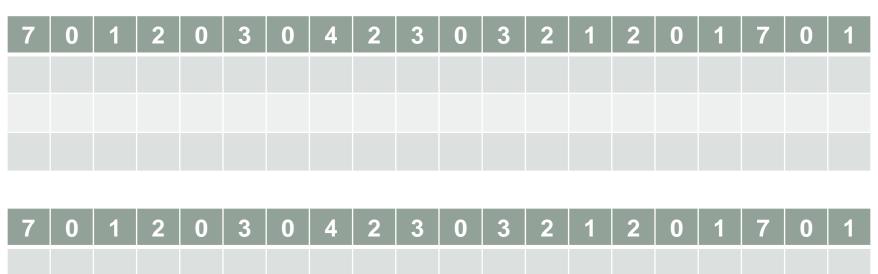
Dr. Matthew Gerber

3/23/2015

### PAGE REPLACEMENT EXAMPLES

#### **Optimal**

Consider references 7,0,1,2,0,3,0,4,2,3,0,3,2,1,2,0,1,7,0,1



#### **Optimal**

Consider references 7,0,1,2,0,3,0,4,2,3,0,3,2,1,2,0,1,7,0,1

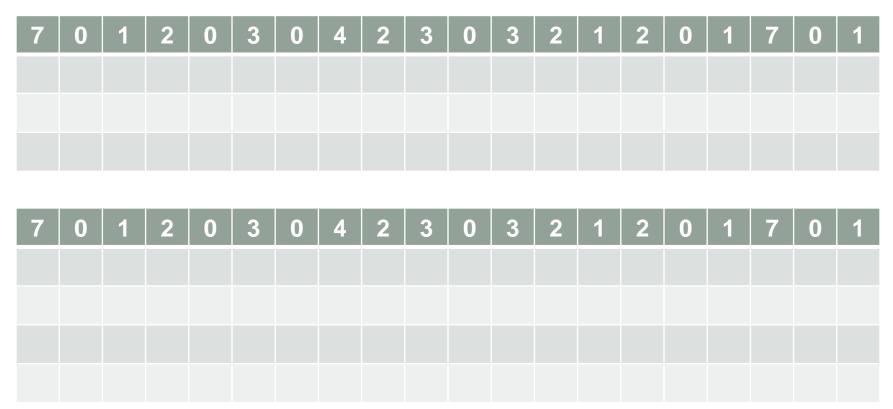
7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	2	2	2	2	2	2	2	2	2	2	2	2	2	2	7	7	7
	0	0	0	0	0	0	4	4	4	0	0	0	0	0	0	0	0	0	0
		1	1	1	3	3	3	3	3	3	3	3	1	1	1	1	1	1	1

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	7	7	3	3	3	3	3	3	3	3	1	1	1	1	1	1	1
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1	1	1	1	1	4	4	4	4	4	4	4	4	4	4	7	7	7
			2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2

• 9 faults for 3 frames, 8 faults for 4 frames

#### First-In, First-Out

Consider references 7,0,1,2,0,3,0,4,2,3,0,3,2,1,2,0,1,7,0,1



#### First-In, First-Out

Consider references 7,0,1,2,0,3,0,4,2,3,0,3,2,1,2,0,1,7,0,1

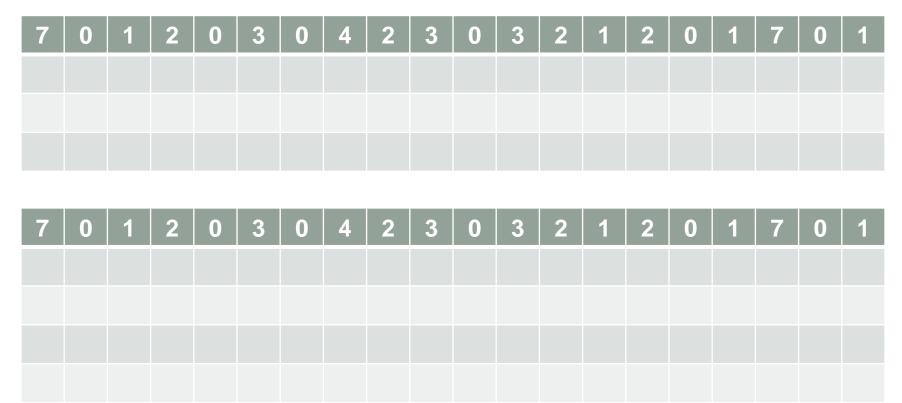
7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	2	2	2	2	4	4	4	0	0	0	0	0	0	0	7	7	7
	0	0	0	0	3	3	3	2	2	2	2	2	1	1	1	1	1	0	0
		1	1	1	1	0	0	0	3	3	3	3	3	2	2	2	2	2	1

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	7	7	3	3	3	3	3	3	3	3	3	2	2	2	2	2	2
	0	0	0	0	0	0	4	4	4	4	4	4	4	4	4	4	7	7	7
		1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
			2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1

• 15 faults for 3 frames, 10 faults for 4 frames (optimal 9, 8)

#### Least Recently Used

Consider references 7,0,1,2,0,3,0,4,2,3,0,3,2,1,2,0,1,7,0,1



#### Least Recently Used

Consider references 7,0,1,2,0,3,0,4,2,3,0,3,2,1,2,0,1,7,0,1

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	2	2	2	2	4	4	4	0	0	0	1	1	1	1	1	1	1
	0	0	0	0	0	0	0	0	3	3	3	3	3	3	0	0	0	0	0
		1	1	1	3	3	3	2	2	2	2	2	2	2	2	2	7	7	7
7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	7	7	3	3	3	3	3	3	3	3	3	3	3	3	7	7	7
7	7 0	7 0	<b>2</b> 7 0									_	1 3 0			1 3 0	7 7 0		1 7 0
7		1 7 0 1		7	3	3	3	3	3	3	3	3		3	3			7	-

• 12 faults for 3 frames, 8 faults for 4 frames (optimal 9, 8)

## FRAME ALLOCATION ALGORITHMS

#### Frame Allocation (9.5)

- By definition we have a certain number of frames available to allocate to user processes
- Given the number of processes and their pattern of memory use, how do we decide how many frames to allocate each process?
- This decision is made by a frame allocation algorithm
- For a given processor we need a certain minimum number of frames per process
  - Less of a concern than it used to be

#### Simple Algorithms (9.5.2-9.5.3)

- Equal allocation simply divides the number of pages by the number of processes
  - Easy to do but doesn't take into account that different processes are different sizes
- Proportional allocation does the same thing but weights by the comparative sizes of the processes
  - Less silly, but still a static approach

#### The Working Set Model (9.6.2)

- Recall the observation that a process has a working set of pages that it is actively using at any given time
- This observation can be translated directly into an approach for frame allocation
- Observe the breadth of a process's memory references and allocate a number of frames equal to the number of pages it has referenced in the past d references
- Consider references:

2 6 1 5 6 6 6 6 5 1 6 2 3 4 1 2 3 4

• Step through with d = 4, d = 5 and d = 6

#### Page Fault Frequency (9.6.3)

- Working Set model works, but is very indirect
- Recall that what we're trying to prevent is thrashing
- Page-fault frequency method observes page faults directly
- If the page fault frequency rises above a certain level, the process is allocated more memory
- If the page fault frequency falls below a certain level, the process may be allocated less memory
  - More likely a process will only be allocated less memory based on external factors

### NEXT TIME: EXAM REVIEW