CSCI-3753: Operating Systems
Fall 2019

Abigail Fernandes

Department of Computer Science

University of Colorado Boulder

Week 12
> Frame Allocation
> Working Set Model

Global vs Local Page Replacement

	Local Replacement	**Global Replacement**
Victim selection	From the own set of allocated frames of each process	From the set of all page frames, even if the page frame is currently allocated to another process
# allocated frames	Fixed	Variable
Control of page fault rate	Affected by the paging behavior set for the process	Cannot be controlled by the process itself
Throughput	Low	High

Frame Allocation

- Each process needs a minimum number of pages.
- Allocation scheme
  - Equal vs Proportional allocation
- If a process is not allocated with enough pages, the page fault rate is very high.
- **Thrashing ->** A process is busy swapping pages in and out.

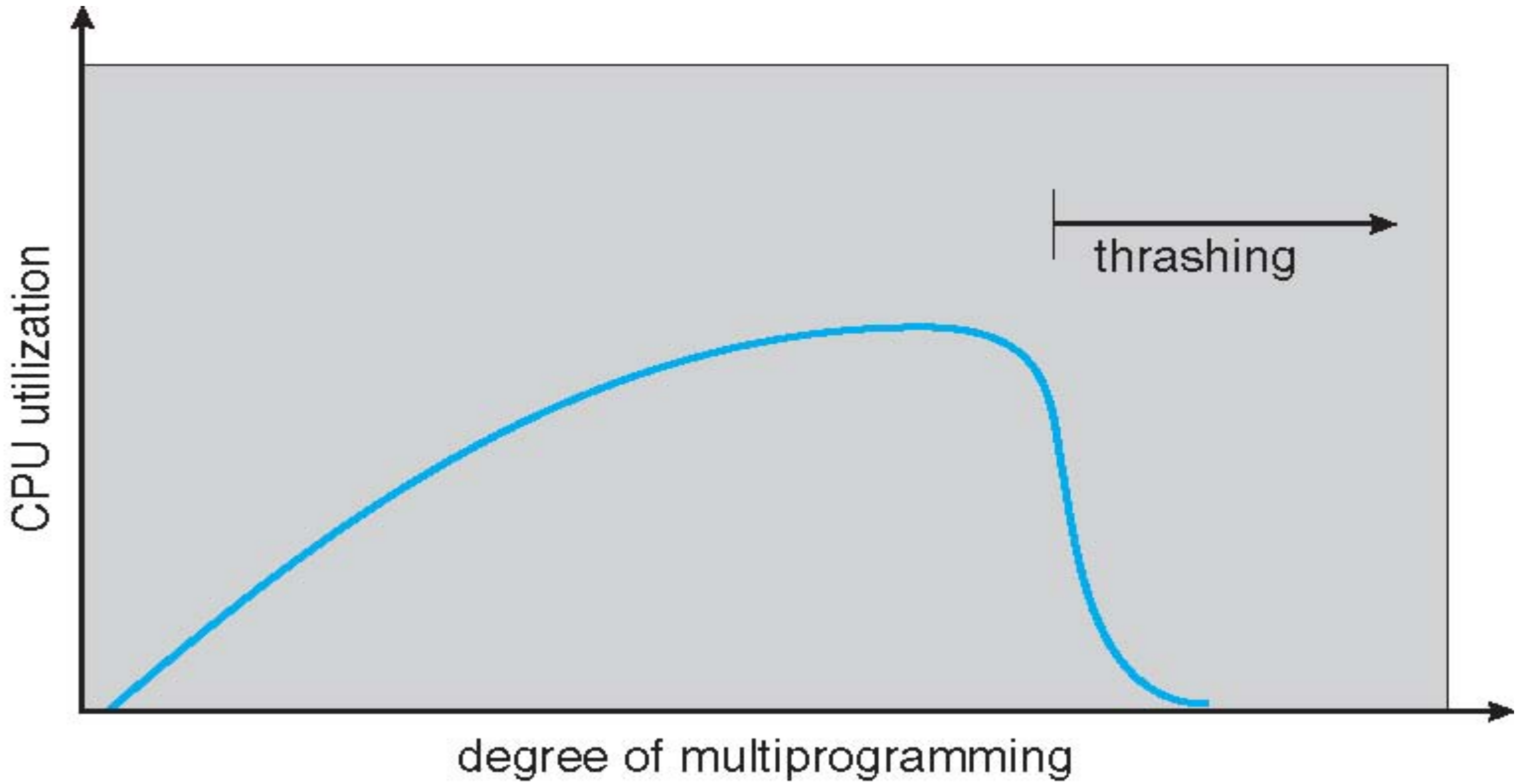
Thrashing

- If a process does not have "enough" pages, the page-fault rate is very high
  - Page fault to get page
  - Replace existing frame
  - But quickly need replaced frame back

High page fault rate leads to
  - Low CPU utilization
  - OS thinking that it needs to increase the degree of multiprogramming
  - Another process added to the system

- **Thrashing** ≡ a process is busy swapping pages in and out

Thrashing



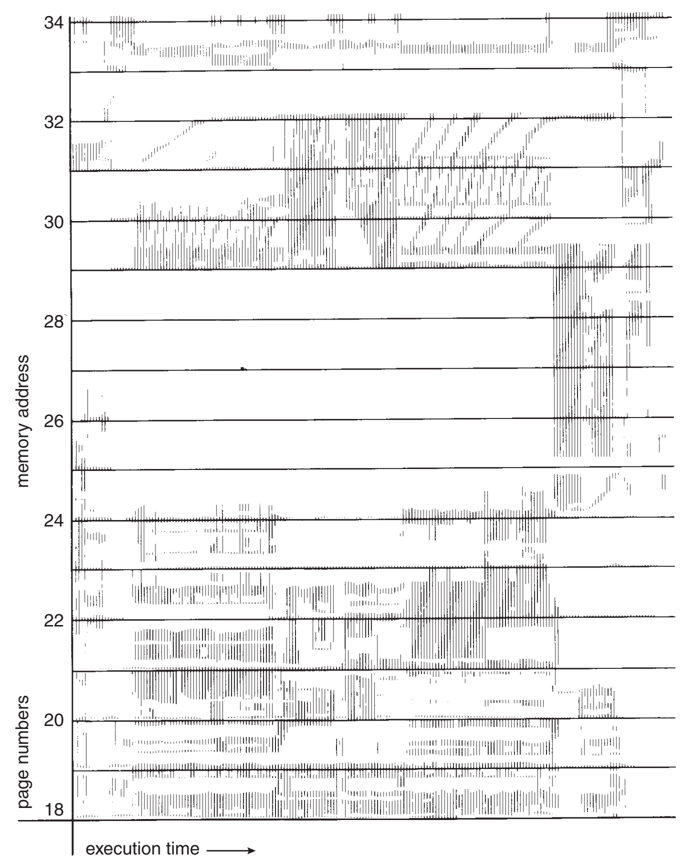
Dealing with Thrashing

- Approach 1: Working set
  - How much memory does the process need in order to make reasonable progress (its working set)?
  - Only run processes whose memory requirements can be satisfied
- Approach 2: Page Fault Frequency
  - PFF = page faults / instructions executed
  - If PFF rises above threshold, process needs more memory
    - Not enough memory on the system? Swap out.
  - If PFF sinks below threshold, memory can be taken away

Week 11
> Frame Allocation
> Working Set Model

Locality in a memory-reference pattern

During any phase of execution, the process references only a relatively small fraction of pages.



Dynamic Page Allocation

- Dynamic Paging uses the locality model – the set of pages used for that particular phase of computation.
- A process migrates from one locality to the next, as the process moves to a different phase of computation.
- Dynamic paging algorithms attempt to adjust the memory allocation to match the process' needs as it executes.

The **Working Set** algorithm is the best known algorithm for dynamic paging.

Approach 1: Working Set Model

- Define Δ to be a working-set window. Δ is a fixed number of page references For example: 10,000 instructions
- WSS_i (working set of Process P_i) is defined to be the total number of pages referenced in the most recent Δ (varies in time)

Working Set Model

Suppose that Δ = 10

Reference string:

5, 5, 101, 5, 5, 5, 101, 5, 5, 5, 100, 5, 103, 6, 6, 6, 6, 6, 6, 6, 7, 7, 7, 7, 7, 7, 103, 7, 101, 7

Δ = 10
WS = {5, 100, 101, 103}

Δ = 10
WS = {6, 7, 101, 103}

Δ = 10
WS = {5, 6, 100, 101, 103}

Working Set Model

- WSS_i -- tries to approximate the size of the locality of process P_i
  - if Δ too small will not encompass entire locality
  - if Δ too large will encompass several localities
  - if $\Delta = \infty \Rightarrow$ will encompass entire program
- $D = \Sigma\ WSS_i \equiv$ approximate the total demand frames
  - Approximation of ALL localities
- m = total number of frames.
- if $D > m \Rightarrow$ Thrashing
- Policy: if D > m, then suspend or swap out one of the processes

Working Set Model

- OS allocates enough page frames to each process to hold its working set.
- If the sum of all cardinality of all working sets for the loaded processes exceeds the number of physical frames, the OS will swap out a process.
- Later, when the process is swapped back into the memory, its working set of pages can be preloaded back into main memory.
- Thus, reducing page faults when the process is restarted by preparing the working set.

Working Set Model

- Reference string

  0, 1, 2, 3, 0, 1, 4, 0, 1, 2, 3, 4
- LRU algorithm
- 3 frames are allocated
- $\Delta = 3$

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

Page fault = 10, Miss ratio = 10/12 = 83.3%, Hit ratio = 16.7%

Working Set Model

- Reference string

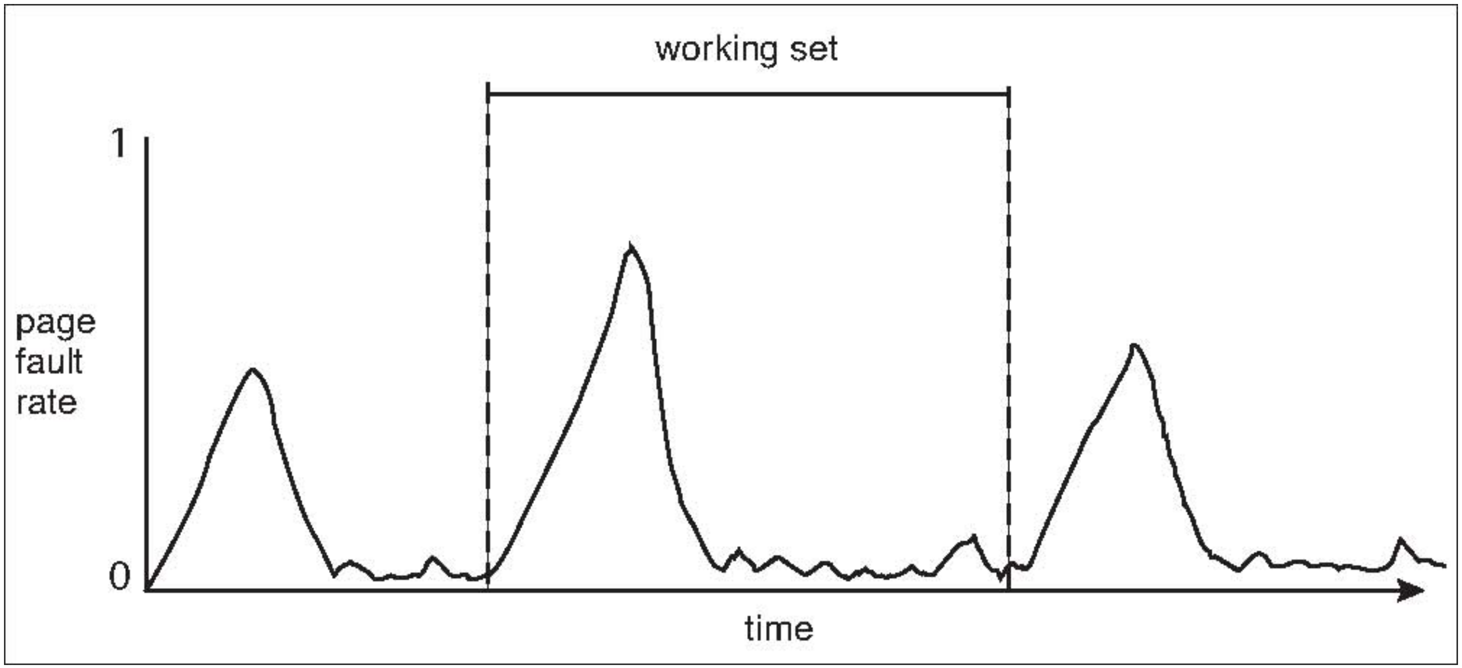
  0, 1, 2, 3, 0, 1, 4, 0, 1, 2, 3, 4

- LRU algorithm
- 4 frames are allocated
- $\Delta = 4$

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

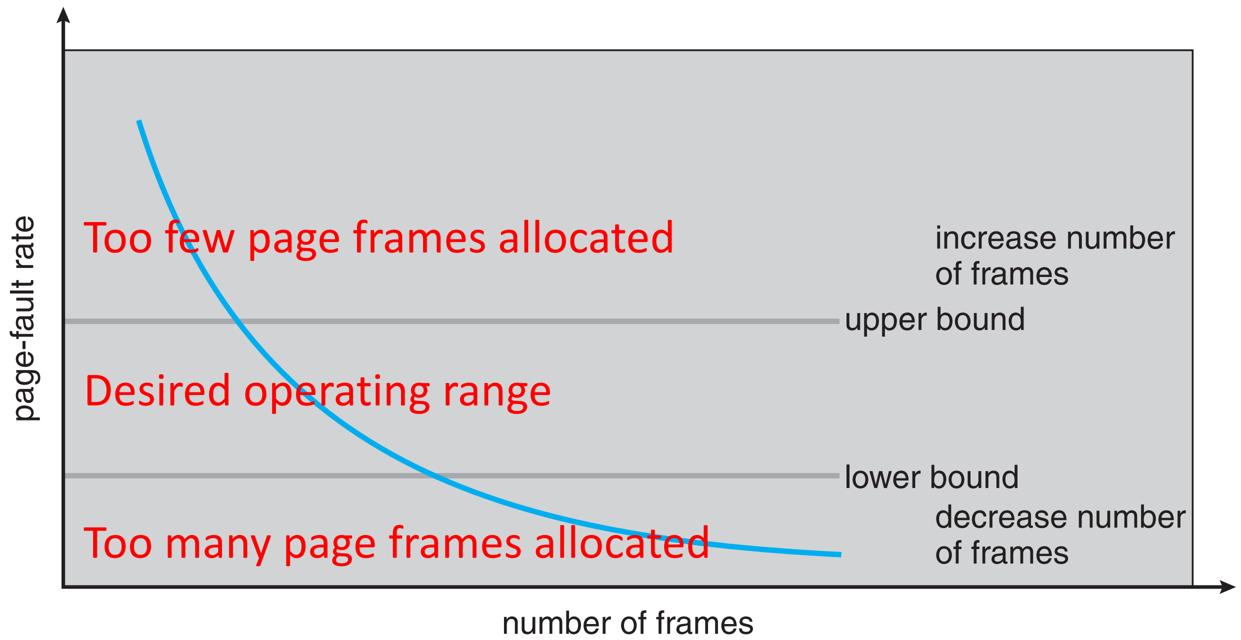
Page fault = 8, Miss ratio = 8/12 = 66.7%, Hit ratio = 33.3%

Working Set vs Page Fault Rate



Approach 2: Page Fault Frequency Scheme

- More direct approach than Working Set Model
- Establish "acceptable" **page-fault frequency** rate and use local replacement policy



Estimating the Working Set

- How do we determine the working set of a process?
- Simple approach
  - Approximate with interval timer + a reference bit
- Example: t = 10,000 instructions
  - Interrupts after every 5000 instructions.
  - Keep in memory 2 bits for each page.
  - Whenever a timer interrupts, shift the bits to right and copy the reference bit value onto the high order bit and sets the values of all reference bits to 0.
  - If one of the bits in memory = 1, page in working set.
- Why is this not completely accurate?
  - Not sure when exactly in the last 5000 time units was this page accessed.
- Improvement = 10 bits and interrupt every 1000 instructions