Paging concepts

Memory resident page

Pages read in on-demand.

Write pages out to storage.

Write “dirty” pages out to disk.

Read in pages from disk on demand.

Keep track of where pages are on disk.
Too little memory $\Rightarrow$ Throwing
- Allocate more physical memory to the process

Plenty of physical memory
$\Rightarrow$ Adding more doesn't help
Reduce process' quota of physical pages
Belady's Anomaly

Reference string: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5

3 frames:

4 frames:

Number of page faults vs number of frames:

- 3 frames: 9 page faults
- 4 frames: 10 page faults
Page replacement comparison

Saturday, December 22, 2018  12:03 PM

First-In-First-Out:
reference string: 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1
page frames: 7 7 7 2 2 2 4 4 4 0 0 0 3 3 3 2 2 2 1 1
                                   1 0 0 3 3 3 2 2 1
15 page faults

Optimal:
reference string: 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1
page frames: 7 7 7 2 2 2 2 2 7
                                   0 4 0 0 0
                                   1 3 3 3 1
9 page faults
(impossible without knowing ref. string in advance)

Least Recently Used:
reference string: 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1
page frames: 7 7 7 2 2 4 4 4 0 1 1 1
                                   0 0 3 3 3 2 2 2 2 2
12 page faults
(require every access to be logged => approximate with 2nd-chance (Used)

Paging Page 4
Using a stack to record page references
Thrashing
Saturday, December 22, 2018  12:05 PM

Useful CPU utilization

Demand for virtual memory

Thrashing starts
Working set model

\[ \Delta \]

\[ WS(t_1) = \{1, 2, 5, 6, 7\} \]

\[ \Delta \]

\[ WS(t_2) = \{3, 4\} \]