Memory Management

- Frame Allocation
- Thrashing and Working Set Model

Frame Allocation
- Fixed number of frames in physical memory
- Varying number of processes:
  - Different memory requirements
  - How to allocate frames to each process?

Bounds on Number of Frames
- What is the minimum number of frames per process?
- Factors to consider:
  - Performance: Want to keep minimum memory required by process
  - Architecture: Some instructions may need to touch multiple pages simultaneously
- What is the maximum number of frames per process?
Frame Allocation Algorithms

- Equal allocation:
  - Each process gets $m/n$ frames for $m$ total frames, $n$ processes
  - Problem?
- Proportional allocation:
  - Each process gets no. frames in proportion to its total size
  - Problem?

Local vs. Global Allocation

- Local Allocation:
  - A process can replace a page from its allocation only
  - Benefit?
  - Problem?
- Global Allocation:
  - A process can replace a page from another process

Thrashing

- Process/system spending more time paging than executing
- How could thrashing occur?
  - Process does not have enough pages for its need
  - Process may borrow pages from other processes causing them to thrash

Preventing Thrashing

- Option 1: Use local page replacement
  - Can this still impact other processes?
- Solution: Provide each resident process enough pages based on its need
  - Working-Set Model
  - Page-Fault Frequency
Locality Principle

- Locality: Set of pages accessed together
  - E.g.: function code, local vars, referenced data
- Process execution: moves between localities
  - E.g.: moves from one function call to another, moves from one part of data to another
- Goal: Allocate enough frames to fit current locality of process

Working Set Model

- Tries to identify the current locality of a process
- Working set: Set of pages in the "most recent" page references
- How do we define "most recent"?
  - Parameter $\Delta$: Window size
  - How to choose $\Delta$?
- Computing working set:
  - Similar to LRU-approximation: Can use reference bit vectors

Working Set Model

- To avoid process thrashing:
  - Assign number of frames = working set size
- To avoid system thrashing:
  - Sum of working sets of all processes $\leq$ total frames
- What if this condition is violated?
  - OS has to suspend and swap out one (or more) process

Page Fault Frequency

- Working set may be difficult to determine. Why?
- What is the symptom of thrashing?
  - High page fault frequency (PFF)
- How is the page fault rate related to working set?
  - High PFF $\Rightarrow$ not enough frames
  - Low PFF $\Rightarrow$ too many frames
Page Fault Frequency

- Thresholds on PFF:
  - High watermark: Increase number of frames
  - Low watermark: Decrease number of frames
- PFF may temporarily increase when:
  - Process changes localities
  - How to handle such temporary spikes?