CSc 360
Operating Systems
Page Allocation

Jianping Pan

Summer 2015

Review

- Page replacement
  - FIFO
  - optimal
  - LRU
  - LRU approximation

Page buffering

- Keep a pool of free pages
  - speed up swapping in desired pages
  - no need to wait a page *becomes* free
- Keep a list of modified pages
  - synch with disk when paging is idle
  - reduce overhead when swapping out
- Reuse “clean” pages from the pool
  - on page fault, check free pool first

Page allocation

- A process needs *minimum* number of pages
  - e.g., IBM 370: 6 pages to handle SS MOVE
    - instruction is 6 bytes, might span 2 pages
    - 2 pages to handle **from**
    - 2 pages to handle **to**
- Two major allocation schemes
  - fixed allocation
  - priority allocation

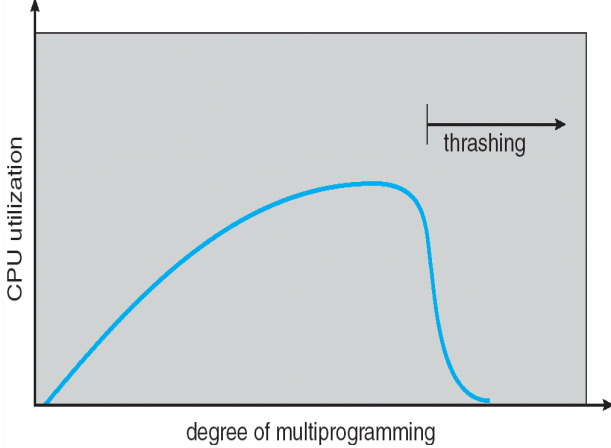
Fixed allocation

- Equal allocation
  - *M* free pages
  - *N* requesting processes
  - allocation: floor(*M*/*N*) each
    - some processes may request less than *M*/*N*
- Proportional allocation
  - each process requests s_i; all request *S*=sum s_i
  - allocation: $s_i/S*M$

Priority allocation

- Allocation proportional to priority
  - higher priority process gets more pages allocated *when* necessary
- On page fault
  - select for replacement one of its frames.
  - select for replacement a frame from a process with lower priority number
- Global vs local replacement

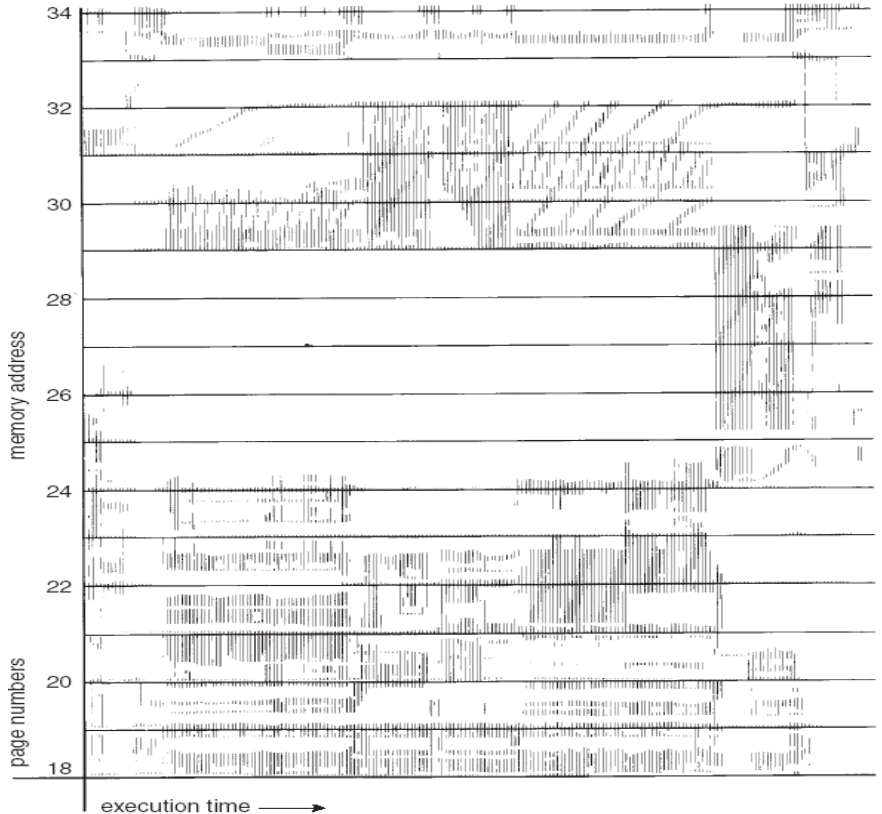
Prevent thrashing



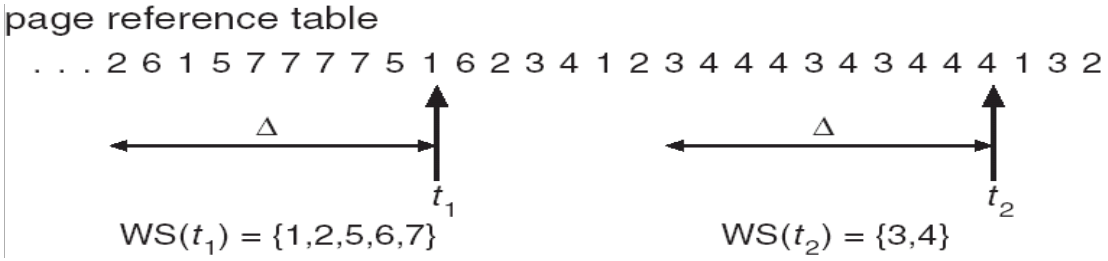
- Thrashing
  - more time on paging than executing
    - busy I/O, idle CPU
    - more processes admitted, more page faults
    - more processes in thrashing!
- Why thrashing
  - paging: explore locality
  - thrashing: locality explored too much!

Paging and thrashing

- Why does thrashing occur?
  - sum of size of locality > total memory size



Working-set model



- Prevent thrashing
  - local replacement
  - sufficient provisioning
  - working-set model
    - working-set window
      - most recent page references
    - working-set size (WSS)
      - number of unique page references
  - when sum $WSS_i > M$, reduce multiprogramming!

This lecture

- Page allocation
  - allocation algorithms
  - thrashing and thrashing prevention
- Explore further
  - OSC7 Section 9.8 and 9.9

Next lecture

- Mass storage
  - read OSC7Ch12