OPERATING SYSTEMS

Memory Management

Suresh Jamadagni
Department of Computer Science

OPERATING SYSTEMS

Virtual Memory

Suresh Jamadagni
Department of Computer Science

OPERATING SYSTEMS

Slides Credits for all the PPTs of this course

- The slides/diagrams in this course are an **adaptation**, **combination**, and **enhancement** of material from the following resources and persons:

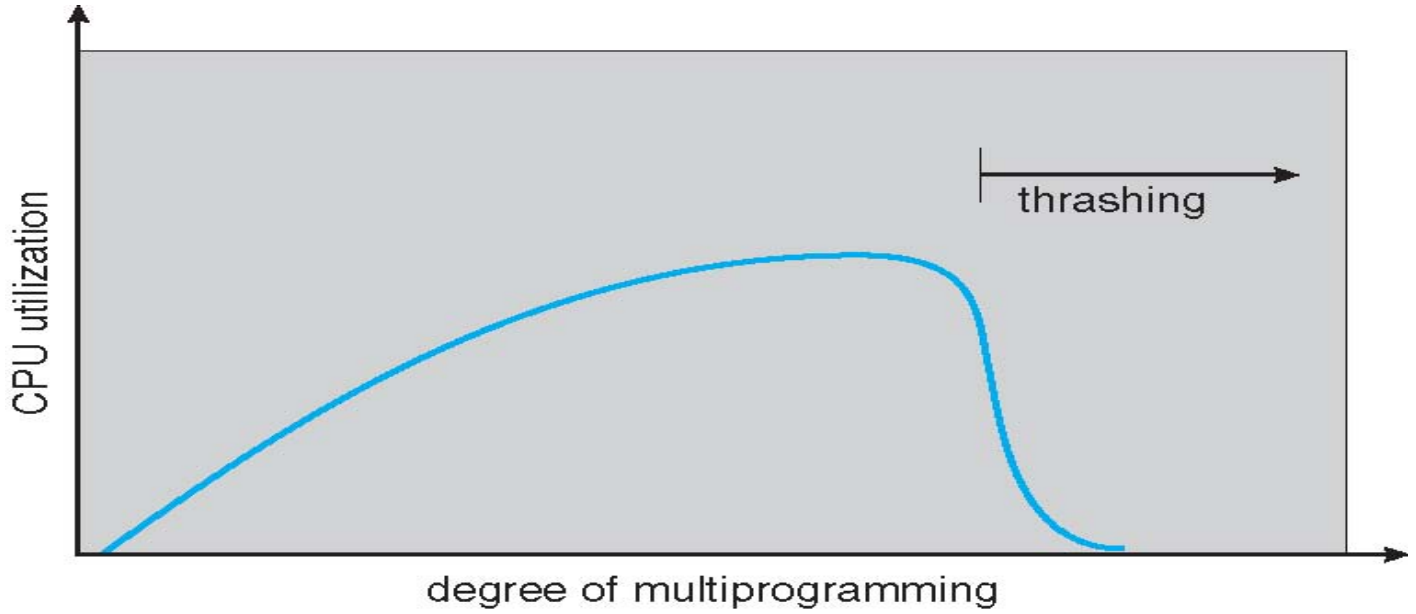
1. Slides of Operating System Concepts, Abraham Silberschatz, Peter Baer Galvin, Greg Gagne - 9th edition 2013 and some slides from 10th edition 2018
2. Some conceptual text and diagram from Operating Systems - Internals and Design Principles, William Stallings, 9th edition 2018
3. Some presentation transcripts from A. Frank – P. Weisberg
4. Some conceptual text from Operating Systems: Three Easy Pieces, Remzi Arpaci-Dusseau, Andrea Arpaci Dusseau

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
  - This leads to:
    - Low CPU utilization
    - Operating system 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

Cause of Thrashing

- Thrashing results in severe performance problems.
- Consider the scenario of the paging systems performance.
- As the degree of multiprogramming increases, CPU utilization also increases.
- Why there is a decrease in the CPU Utilization when the degree of multiprogramming is increased?.
- How this will be handled?

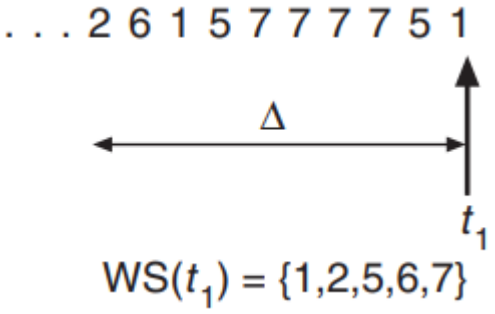


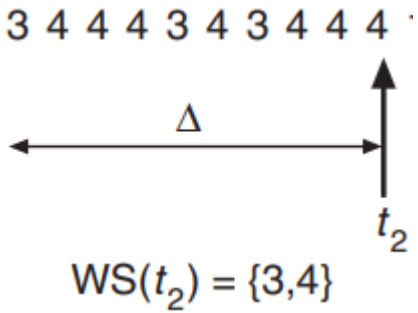
Cause of Thrashing contd…

- We can Limit the effects of thrashing by using a **local replacement algorithm** (or priority replacement algorithm)
  - If process starts thrashing, it cannot steal frames from another processes
- how does demand paging work?
  Locality model
  - Process migrates from one locality to another
  - Localities may overlap
- Why does thrashing occur?
  Σ size of locality > total memory size
  - Limit effects by using local or priority page replacement

Keeping Track of the Working Set

- The working-set model is based on the assumption of locality.
- The parameter, Δ , defines the working-set window.
- It is examine the most recent Δ page references.
- The set of pages in the most recent Δ page references is the working set.
- If a page is in active use, it will be in the working set.
- If it is no longer being used, it will drop from the working set time units after its last reference.
- Thus, the working set is an approximation of the program's locality
- Approximate the **W-S** model with interval timer + a reference bit





Keeping Track of the Working Set

- Example: Δ = 10,000 references
  - Timer interrupts after every 5000 time units
  - Keep in memory 2 bits for each page
  - Whenever a timer interrupts copy and sets the values of all reference bits to 0
  - If one of the bits in memory = 1 $\Rightarrow$ page in working set
- Why is this not completely accurate?
  - We cannot tell where, within an interval of 5000, a reference occurred.
- Improvement = 10 bits and interrupt every 1000 time units but overhead to service more frequent interrupts

OPERATING SYSTEMS

Working-Set Model

- Parameter $\Delta \equiv$ working-set window $\equiv$ a fixed number of page references

  Example: 10,000 instructions

- WSS_i (working set of Process P_i) =

  total number of pages referenced in the most recent Δ (parameter that varies in time)

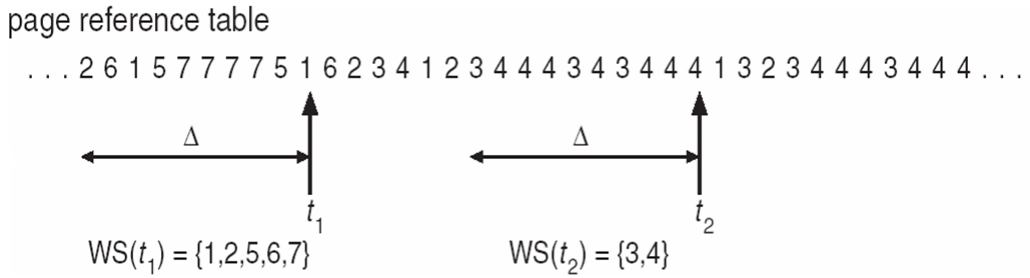
  - if Δ too small will not encompass entire locality
  - if Δ too large will encompass several localities
  - if $\Delta = \infty \Rightarrow$ will encompass entire program

- $D = \Sigma\ WSS_i \equiv$ total demand frames

  - Approximation of locality

- if $D > m$ *(available frames)* $\Rightarrow$ Thrashing will occur

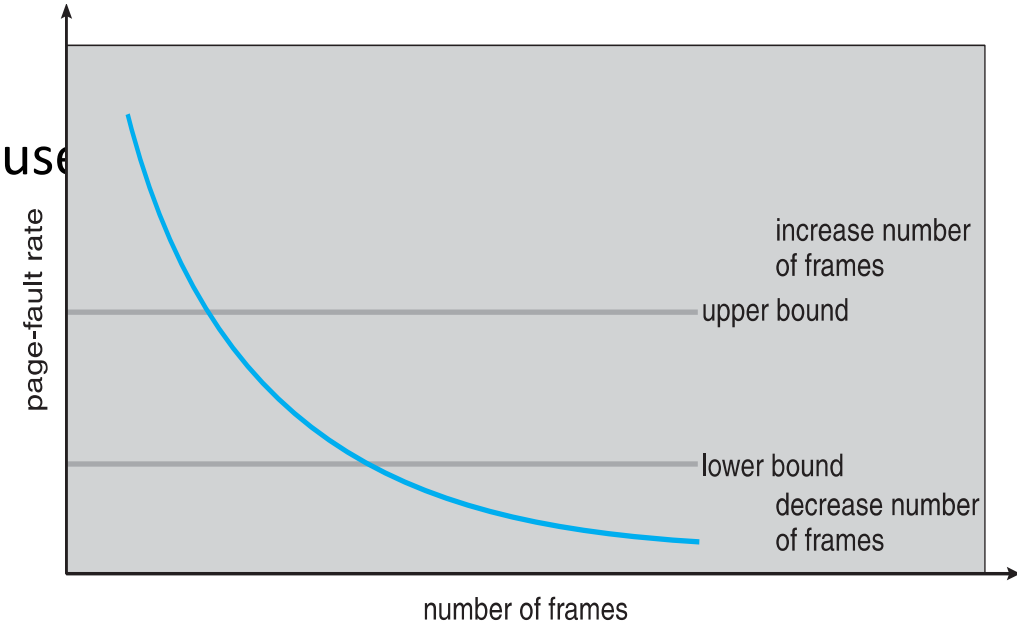
- Policy if $D >$ m, then suspend or swap out one of the processes



OPERATING SYSTEMS

Page-Fault Frequency

- More direct approach than W-S model
  - Control the page-fault rate
  - Looks clumsy to control thrashing
- Establish "acceptable" **page-fault frequency** (**PFF**) rate and use local replacement policy
  - If actual rate too low, process loses frame
  - If actual rate too high, process gains frame.
  - processes can be suspended if the free frames is zero
- With the working-set strategy, we may have to swap out a process.
  - If page fault Increases and no free frames available



THANK YOU

Suresh Jamadagni
Department of Computer Science and Engineering

sureshjamadagni@pes.edu