Virtual Memory II

Allocation of Frames

Limits to Allocations:

- **Maximum:** Maximum frames used by process depends upon total available frames, and frames used by the OS.
- **Minimum:** Minimum frames depend upon system implementation.

Initial Allocations: We can choose how many frames does a process begin with.

NOTE: There are total of *m* frames in the system, and OS uses *k* frames.

- **Equal Allocation:**
  - If there are *n* processes, each one gets equal frames.
  - Each process gets *(m-k)/n* frames to start with.
- **Proportional Allocation:**
  - If s_i is the virtual-memory size of i-th process, then *S = sum(s_i)*
  - Each process gets *a_i = s_i/S * (m-k)* frames

Allocation Management:

- **Local Algorithm:** Each process owns a fixed number of frames.
- **Global Algorithm:**
  - Each process owns a non-constant number of frames dependent on PFF (Page Fault Frequency)
  - Higher the PFF, more the frames owned.

Thrashing

Thrashing Cascade:

- **Global Algorithm:**
  - Assume a system running multiple processes.
  - A process moves out of its locality/locale, causing a high PFF (Page Fault Frequency).
  - High-PFF process takes pages from other processes.
  - *Sometimes,* this will lead to the other process having too few frames, causing it to have a high-PFF too.
  - Overall CPU utilization reduces due to excessive total PFF, CPU runs more processes.
  - New processes require a minimum number of frames - taken from the frames of already-running processes, causing even higher PFF**.**
  - Steady-state of system is Thrashing**.**
- **Local Algorithm:**
  - Assume system running mutiple processes.
  - A process moves out its locale, causing high PFF.
  - DMA is busy, and requests from other process start queue-ing, causing many processes to be blocked for a long duration.
  - Since CPU utilization is low, more processes are launched - making DMA even busier.
  - Cycle repeats, and the system keeps Thrashing.

Preventing Cascade:

- Checking PFF before reducing/increasing number of running processes.
- Give frames to the processes based on the size of their **current** locale.

Working-Set Model

Operation

- Most memory accesses are local.
- A process will have *n* frames allocated to it.
- Last *n* recently used pages will be part of the *working-set.*
    - A page not used since last *n* memory accesses will be removed from *working-set*

Precautions: Too small a value of Working Set will lead to the entire locale *not* being in main-memory