

Virtual Memory

Belady's Anomaly: Increasing the frame allocation increases the number of page faults.

Stack Algorithms:

1. The set of pages kept in memory with an allocation of n frames is a subset of the pages kept in memory for $n+1$ frames.
2. All referenced pages are kept in a data structure called a stack. The top n pages of the stack are the pages kept in memory for n frames. The stack must be handled as follows.
 - a. When a page is referenced it is brought to the top.
 - b. No page below the referenced page's original position can move in any direction.
 - c. Pages above the referenced page's original position can remain in the same place or move down. They may not move up.
 - d. Unreferenced pages are treated as if they were at the bottom of the stack. We make the stack one element larger to accommodate them.

Variable frame allocation

1. Working Set

Working Set Window

2. Page-Fault Frequency

When a page fault occurs, add a page to the allocation

If it has been more than K units of time since the last page fault, remove all unreferenced pages from the allocation.

This algorithm suffers from a form of Belady's anomaly. Increasing K can increase the number of page faults for a particular string.

Implementation of algorithms.

FIFO and PFF – These can be implemented exactly as described because all computation is done when a page fault occurs.

LRU – Can be implemented precisely if additional hardware is used. Is usually implemented as an approximation.

1. At the hardware level we add reference and dirty bits to the page table entry. The reference bit is set whenever a page is referenced (May be inexact) The dirty bit is set any time a value is stored into the page.
2. Implement a series of timer interrupts, and count them. Every time an interrupt occurs, reset all reference bits. If the reference bit was set for a page, use the current count of timer interrupts as the time the page was last referenced.
3. This procedure will not give a unique least recently used page. In the case of duplicates, no n-dirty pages are preferred.

WS – Also implemented using timer interrupts.

Pages are “aged” based on the number of timer interrupts that have occurred since the last reference. Pages beyond a certain age are removed from the allocation. (Sometimes called “The Working Set.”)

Second-Chance

A variant of FIFO.

Reset all reference bits when a page fault occurs.

If a page has been referenced since the last fault, then we ignore it and go on to the next page in the list.

If all pages have been referenced, we choose the oldest. (reverts to FIFO).

LFU/MFU – count references

Page buffering – keep a pool of free frames. When a page is stolen it goes to the page buffer. If it is referenced again right away, it is returned to the process. No I/O is necessary, although a page fault occurs.

Global/Local allocation of frames.

Global – process cannot control its own page fault rate. (When do you do this anyway?)

Local – Frames may be assigned to processes that aren't using them.

Inverted page tables

I/O interlock

TLB-Reach