Page Buffering

LRU and the Clock Algorithm are generally superior to FIFO, but they are more complex and involve more overhead.

Also, it costs more to replace a modified page than an unmodified page.

Page Buffering

- basic strategy is FIFO, but…
- page table entry is added to one of two lists:
  - free page list, if page has not been modified
  - modified page list, otherwise
- try to keep at least a few frames free at all times
- pages are not removed from memory until they MUST be overwritten
- modified pages are replaced only if there is no free frame
- used in VAX/VMS

Resident Set Size

How many frames should be allocated to a process?

Fixed-allocation

- Gives a process a fixed number of pages within which to execute
- When a page fault occurs, one of the pages of that process must be replaced

Variable-allocation

- Number of pages allocated to a process varies over the lifetime of the process

Fixed Allocation, Local Scope

Decide ahead of time the amount of allocation to give a process

If allocation is too small, there will be a high page fault rate

If allocation is too large there will be too few programs in main memory

Variable Allocation, Global Scope

Easiest to implement

Adopted by many operating systems

Operating system keeps list of free frames

Free frame is added to resident set of process when a page fault occurs

If no free frame, replaces one from another process

Variable Allocation, Local Scope

When new process added, allocate number of page frames based on application type, program request, or other criteria

When page fault occurs, select page from among the resident set of the process that suffers the fault

Reevaluate allocation from time to time

Cleaning Policy

Demand cleaning

- A page is written out only when it has been selected for replacement

Precleaning

- Pages are written out in batches

Best approach uses page buffering

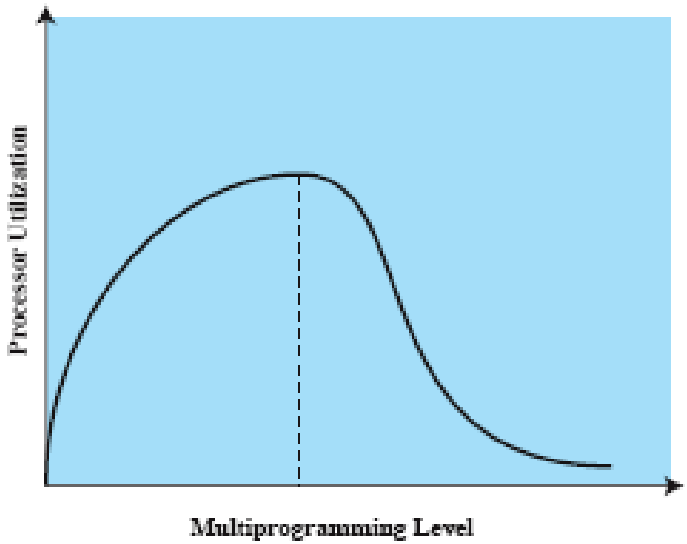
- Replaced pages are placed in two lists
  - Modified and unmodified
- Pages in the modified list are periodically written out in batches
- Pages in the unmodified list are either reclaimed if referenced again or lost when its frame is assigned to another page

Load Control

Determines the number of processes that will be resident in main memory

Too few processes, many occasions when all processes will be blocked and much time will be spent in swapping

Too many processes will lead to thrashing



Process Suspension

Lowest priority process

Faulting process

- This process does not have its working set in main memory so it will be blocked anyway

Last process activated

- This process is least likely to have its working set resident

Process with smallest resident set

- This process requires the least future effort to reload

Largest process

- Obtains the most free frames

Process with the largest remaining execution window

Linux Memory Management

Page directory

Page middle directory

Page table