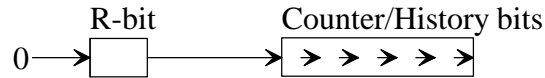


Absent:

1. To approximate the LRU page-replacement algorithm most hardware supports the updating/setting of a reference bit (R-bit) in the page-table entry corresponding to each memory reference. To get a better approximation of LRU, additional counter/history bits can be maintained. Periodically, say every 20 milliseconds, the process is interrupted so the OS can shift the R-bit into the counter/history bits and clear the R-bit. Such as



a) If the pages in main memory have the below R-bit and counter/history bits, then which page should be selected for replacement on a page fault?

Page	R-bit	Counter/History bits
0	0	0 1 0 1 0 1 0
1	1	0 0 1 0 0 0 0
2	1	0 0 0 1 1 1 1
3	0	0 0 1 1 0 1 0
4	0	1 1 0 1 0 1 0

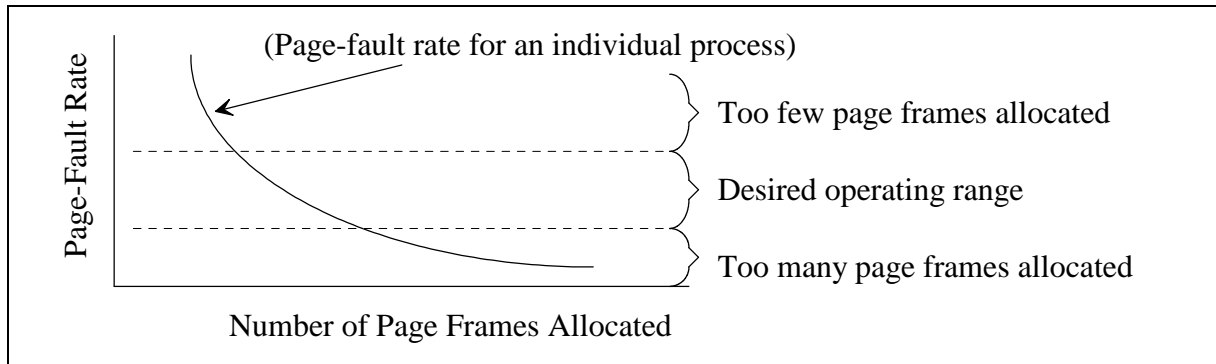
b) If the R-bits are shifted every 20 milliseconds and R-bits are about to be shifted because the interrupt just occurred, how long (specify a range) has it been since page 0 was referenced?

c) If a file read occurred during the middle of a 20 millisecond interval, the OS will switch to running another process. What should the OS do so the page-replacement algorithm can be resumed when the I/O completes and the process starts executing again?

2. Systems using global-page replacement tend to have more problems with thrashing. Why do you suppose this is?

Absent:

3. In the page-fault-frequency frame-allocation algorithm, the OS monitors the page-fault rate of each process to decide if it has too many or not enough page frames allocated to it.



a) If the page-fault rate for a process is too high and the free-frame list is empty, what should the OS do?

b) How could the OS determine the page-fault rate for a process?

c) Do you see any problems with the page-fault frequency approach for allocating page frames?