

Least Recently Used (LRU). Page Replacement algorithm

In Operating systems that utilization of paging for memory management, page replacement algorithms are expected to conclude which page should be replaced when new page comes in. Whenever another page comes in and not present in memory, page fault happens and Operating System replaces one of the current pages with recently required page.

Different page replacement algorithms recommend various types of concluding which page to be replaced. The objective for all algorithm is to decrease the number of page faults.

In Least Recently Used (LRU) algorithm is a Greedy algorithm where the page to be replaced is least recently used. Preference of the idea is based on the locality of the references; the least used page is not likely.

			2	2	2	2	2	2	2	2	2	2
		1	1	1	1	1	4	4	4	4	4	4
	0	0	0	0	0	0	0	0	0	0	0	0
7	7	7	7	7	3	3	3	3	3	3	3	3

Let's take page reference string is 7 0 1 2 0 3 0 4 2 3 0 3 2.

Initially all slots are empty, so when 7 0 1 2 are allocated to the empty slots.
(4 page fault)

After we allocate 0, but 0 is already present in memory. (page hit)

After we allocate 3, it will take the place of 7 due to LRU. (1 page fault)

After we allocate 0. It is already present in memory (page hit).

After we allocate 4. It will take the place of 1 due to LRU. (1 page fault)

After we allocate 2. It is already present in memory. (page hit)

After we allocate 3. It is already present in memory. (page hit)

After we allocate 0. It is already present in memory. (page hit)

After we allocate 3. It is already present in memory. (page hit)

After we allocate 2. It is already present in memory. (page hit)

So, there is total 6 page faults.