

## **Explanation of Working of LFU Page Replacement Algorithm**

The Least Frequently Used (LFU) page replacement algorithm manages cache memory by keeping track of how often each page is accessed. At the beginning of the process, the cache is empty. The cache size and the page reference sequence are taken as input. Along with this, frequency counters for each page are initialized to zero, and counters for hits and misses are also initialized.

The algorithm then processes the page reference sequence one page at a time. For every page access, the algorithm first checks whether the page is already present in the cache. If the page is found, it is treated as a hit. In this case, the frequency count of the accessed page is increased by one, indicating that the page is being used frequently. No replacement occurs during a hit, and the algorithm moves on to the next page reference.

If the requested page is not found in the cache, a miss (page fault) occurs. The miss counter is incremented, and the algorithm checks whether the cache has any free space. If the cache is not full, the new page is simply inserted into the cache, and its frequency count is set to one, since it has been accessed for the first time.

When a miss occurs and the cache is already full, the LFU replacement policy is applied. The algorithm identifies the page in the cache with the lowest frequency count, as this page has been accessed the least number of times. This least frequently used page is removed from the cache, and the new page is inserted in its place with its frequency initialized to one.

After each page access—whether it results in a hit or a miss—the algorithm proceeds to the next page in the reference sequence. This process continues until all page references have been handled. Once the sequence is complete, the total number of hits and misses is calculated, and the hit ratio is computed to evaluate the performance of the cache. Finally, the results are displayed, and the algorithm terminates.