Unit: III
Lecture: 8
(Memory Management)
Page Replacement Algorithms

Page Replacement Algorithms:

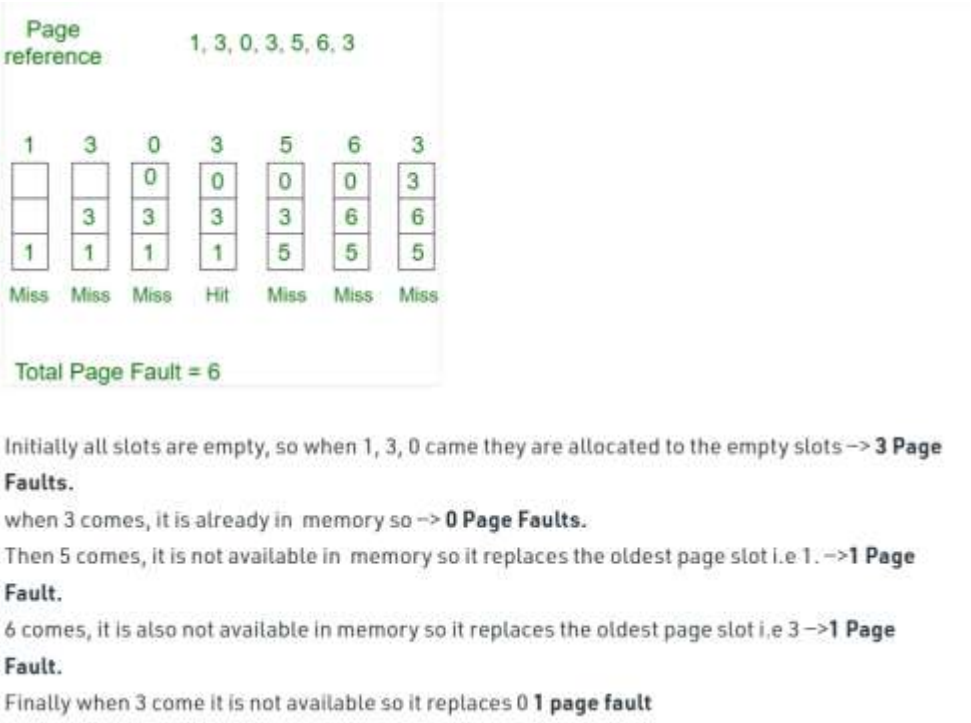
A page fault happens when a running program accesses a memory page that is mapped into the virtual address space, but not loaded in physical memory.

Since actual physical memory is much smaller than virtual memory, page faults happen. In case of page fault, Operating System might have to replace one of the existing pages with the newly needed page. Different page replacement algorithms suggest different ways to decide which page to replace. The target for all algorithms is to reduce the number of page faults.

- **First In First Out (FIFO):** This is the simplest page replacement algorithm.
    - The operating system keeps track of all pages in the memory in a queue, the oldest page is in front of the queue. When a page needs to be replaced, page in the front of the queue is selected for removal.

"Page frame is the smallest fixed-length contiguous block of physical memory into which memory pages are mapped by the operating system."

Example: Consider page reference string 1,3,0,3,5,6 with 3 page frames. Find the number of faults.

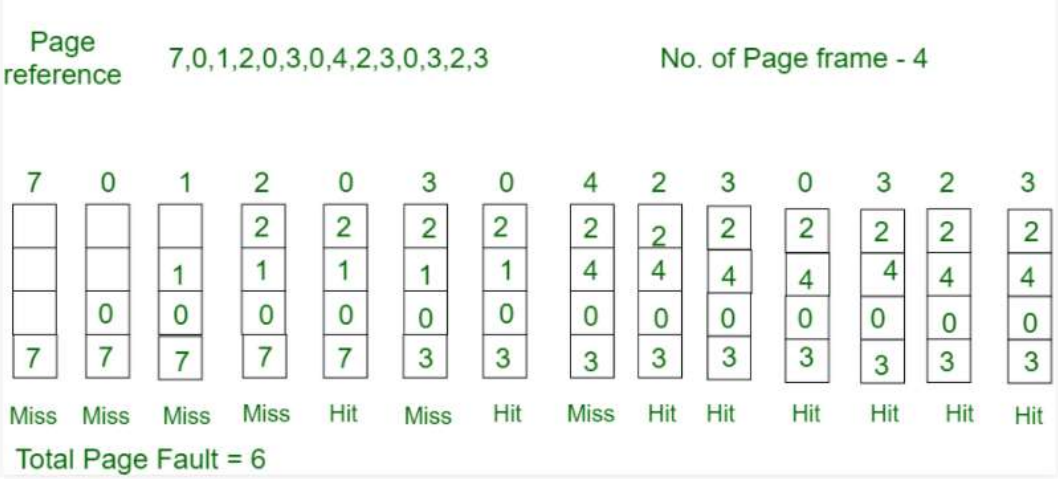


Initially all slots are empty, so when 1, 3, 0 came they are allocated to the empty slots –> **3 Page Faults.**
when 3 comes, it is already in memory so –> **0 Page Faults.**
Then 5 comes, it is not available in memory so it replaces the oldest page slot i.e 1. –>**1 Page Fault.**
6 comes, it is also not available in memory so it replaces the oldest page slot i.e 3 –>**1 Page Fault.**
Finally when 3 come it is not available so it replaces 0 **1 page fault**

Belady's anomaly: Belady's anomaly proves that it is possible to have more page faults when increasing the number of page frames while using the FIFO page replacement algorithm. e.g. if we consider reference string 3,2,1,0,3,2,4,3,2,1,0,4 and 3 slots, we get 9 total page faults, but if we increase slots to 4, we get 10 page faults.

- **Optimal Page Replacement:** In this algorithm, pages are replaced which would not be used for the longest duration of time in the future.

Example: Consider the page references 7,0,1,2,0,3,0,4,2,3,0,3,2 with 4 page frames:



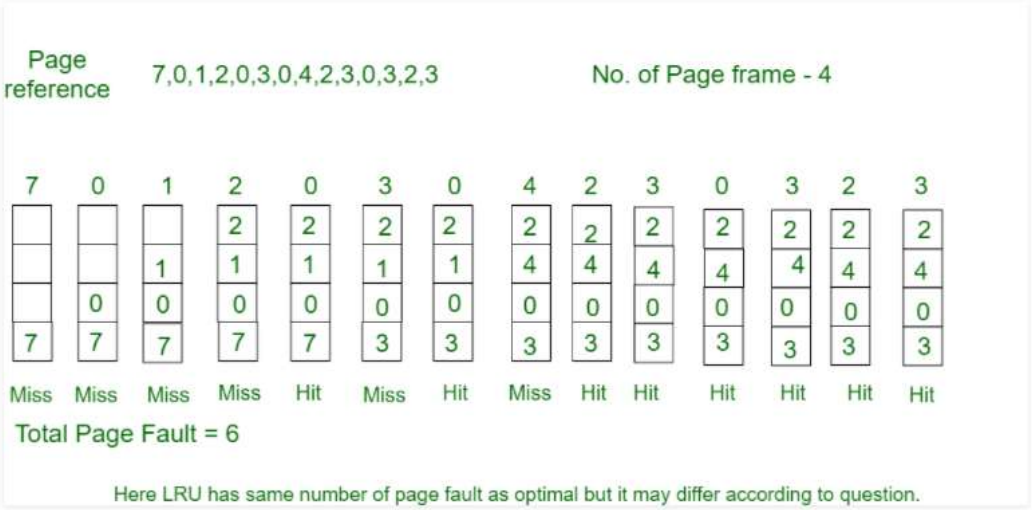
Initially all slots are empty, so when 7 0 1 2 are allocated to the empty slots –> **4 Page faults**
0 is already there so –> **0 Page fault.**
when 3 came it will take the place of 7 because it is not used for the longest duration of time in the future.–>**1 Page fault.**
0 is already there so –> **0 Page fault.**.
4 will takes place of 1 –> **1 Page Fault.**

Now for the further page reference string –> **0 Page fault** because they are already available in the memory.

Optimal page replacement is perfect, but not possible in practice as the operating system cannot know future requests. The use of Optimal Page replacement is to setup benchmark so that other replacement algorithms can be analyzed against it.

- **Least Recently Used:** In this algorithm, page will be replaced which is least recently used.

Example: Consider the page reference string 7,0,1,2,0,3,0,4,2,3,0,3,2 with 4 page frames.



Initially all slots are empty, so when 7 0 1 2 are allocated to the empty slots –> **4 Page faults**

0 is already their so –> **0 Page fault.**

when 3 came it will take the place of 7 because it is least recently used –>**1 Page fault**

0 is already in memory so –> **0 Page fault**.

4 will takes place of 1 –> **1 Page Fault**

Now for the further page reference string –> **0 Page fault** because they are already available in the memory.