TECHNO INTERNATIONAL
BATANAGAR

TERM PAPER ON LRU PAGE REPLACEMENT TECHNIQUE

NAME : SURABHI

ROLL NO. : 33200118008

STREAM : C.S.E.

SEMESTER : 5TH

SUBJECT : OPERATING SYSTEM

SUBJECT CODE : PCC-CS-502

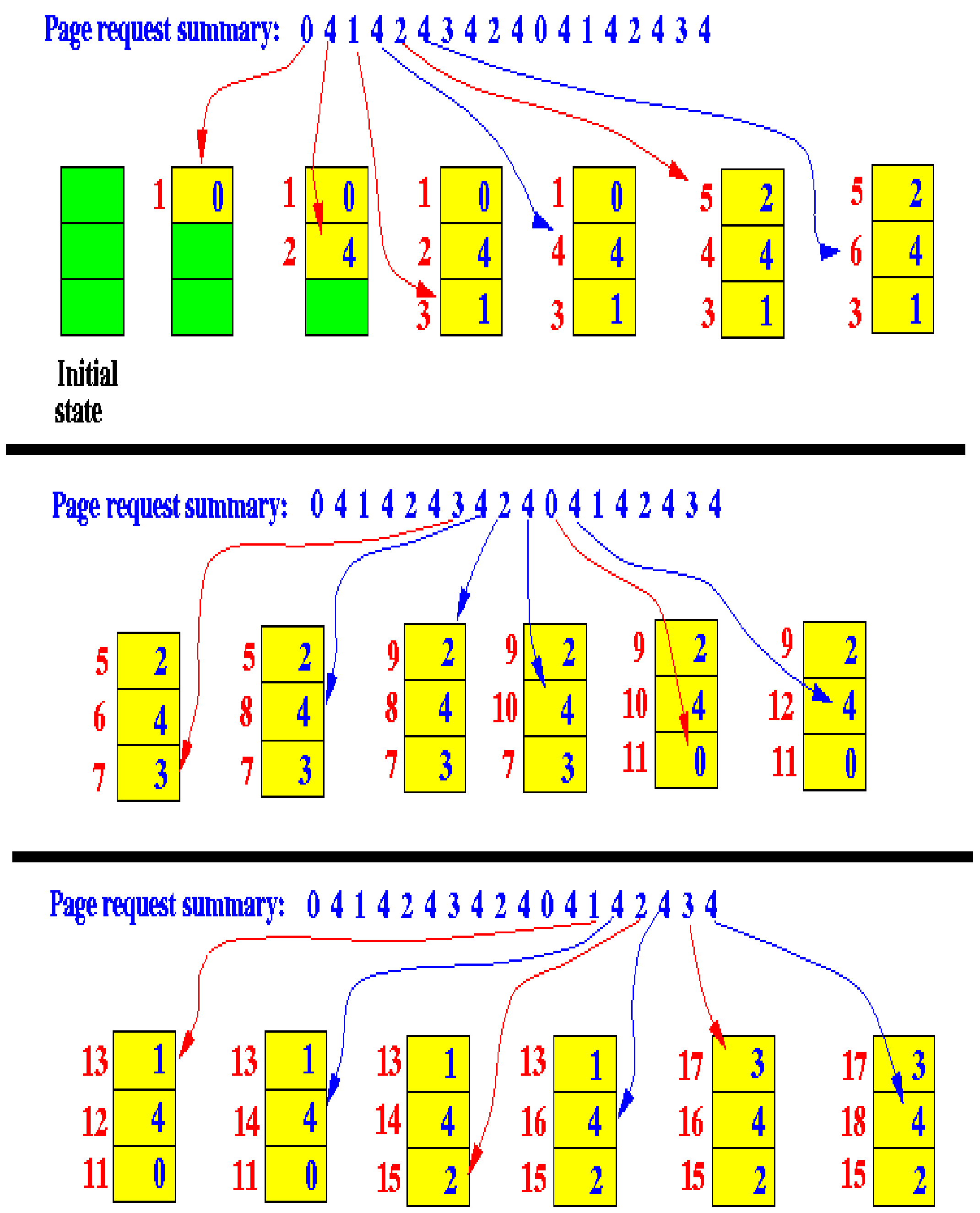
In an operating system that uses paging for memory management, a page replacement algorithm is needed to decide which page needs to be replaced when new page comes in.

Page Fault – 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.

The LRU Page Replacement Policy

- In the Least Recently Used (LRU) page replacement policy, the page that is used least recently will be replaced.
- Implementation:
    - Add a register to every page frame - contain the last time that the page in that frame was accessed
    - Use a "logical clock" that advance by 1 tick each time a memory reference is made.
    - Each time a page is referenced, update its register

- The following figure shows the behavior of the program in paging using the LRU page replacement policy :


Page request summary: 0 4 1 4 2 4 3 4 2 4 0 4 1 4 2 4 3 4
Initial state
Page request summary: 0 4 1 4 2 4 3 4 2 4 0 4 1 4 2 4 3 4
Page request summary: 0 4 1 4 2 4 3 4 2 4 0 4 1 4 2 4 3 4

- We can see notably that the bad replacement decisions made by FIFO is not present in LRU.
  - There are a total of 9 page read operations to satisfy the total of 18 page requests - that is almost a 20% improvement over FIFO in such a short experiment
  - (I only want to make the point here that page replacement policy can affect the system performance. I do not want to get into the question of "how much better is LRU than FIFO").
- In fact, it has been shown empirically that LRU is the preferred page replacement policy
- Disadvantage of LRU replacement policy:
  - To identify the page to replace, you need to find the minimum time stamp value in all the registers...

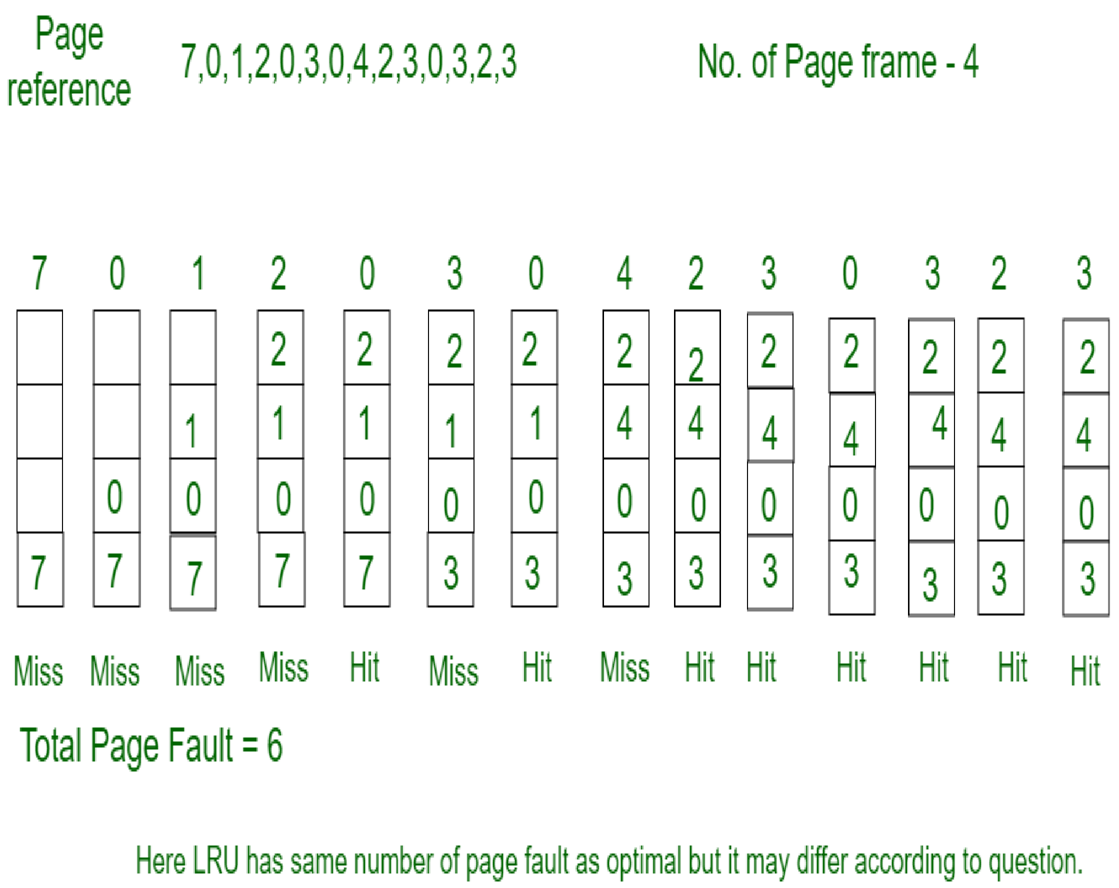
    Lots of work...

- There is a more efficient scheme that approximate the behavior of LRU that runs more efficiently...

That method is "second chance" - next.

EXAMPLE :

Consider the page reference string 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2 with 4 page frames.Find number of page faults.


Page reference
7,0,1,2,0,3,0,4,2,3,0,3,2,3
No. of Page frame - 4
7 0 1 2 0 3 0 4 2 3 0 3 2 3
Miss Miss Miss Miss Hit Miss Hit Miss Hit Hit Hit Hit Hit Hit
Total Page Fault = 6
Here LRU has same number of page fault as optimal but it may differ according to question.

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.