R V COLLEGE OF ENGINEERING

EXPERIENCE LEARNING

NAME	SHIVARAJ CHAWAN , VISHWANATH A D
USN	1RV2CS185, 1RV22CS2234
CLASS	CSE-D
TOPIC	IMPLEMENTING FIFO USING SECOND CHANCE PAGING POLICY

PAGING CONCEPT

- **The process of retrieving processes in the form of pages from the secondary storage into the main memory is known as paging.**

- **Basic terms used in paging are page offset, paging table,logical address space, physical address space,frame size.**

- **These are some advantages of using paging:**

  1.Paging allows for efficient use of physical memory by dividing it into fixed-size blocks (page frames) and virtual memory into corresponding pages.

  2.The operating system can easily allocate and deallocate memory pages without the need for contiguous physical memory blocks

  .

  3.Unlike some memory allocation strategies, such as contiguous memory allocation, paging eliminates external fragmentation.

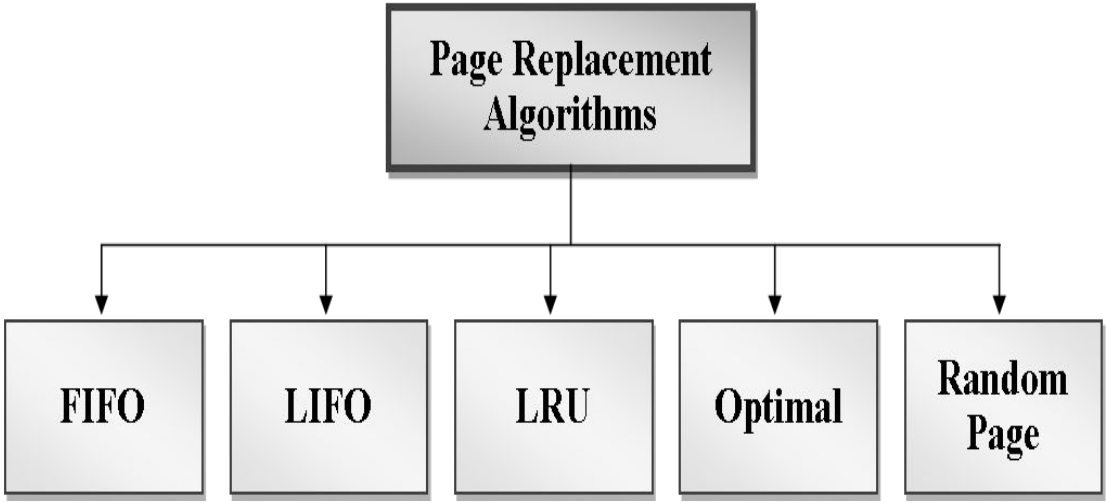
  4Paging facilitates efficient multitasking by allowing the operating system to quickly switch between different processes. Each process can have its own set of pages, and the operating

4. Paging facilitates efficient multitasking by allowing the operating system to quickly switch between different processes. Each process can have its own set of pages, and the operating system can easily manage their execution.

5. Page tables can be used to implement memory protection mechanisms. Each page can be assigned specific access permissions (read-only, read-write, execute-only, etc.), providing a level of security and preventing unauthorized access.

Here are some paging replacement policies:

1. Optimal Page Replacement.
2. FIFO (First-In-First-Out)
3. LRU (Least Recently Used)
4. RANDOM PAGE REPLACEMENT
5. LIFO(LAST IN FIRST OUT)



FIFO(FIRST IN FIRST OUT)

FIFO, which stands for First-In-First-Out, is a simple and straightforward page replacement algorithm used in operating systems. It operates on the principle of replacing the oldest page in memory when a page fault occurs. Here's how FIFO page replacement works:

1.In these we need to know how many frames is assigned and now we should go the reference String which consist set of pages of processes should be added in the frames.

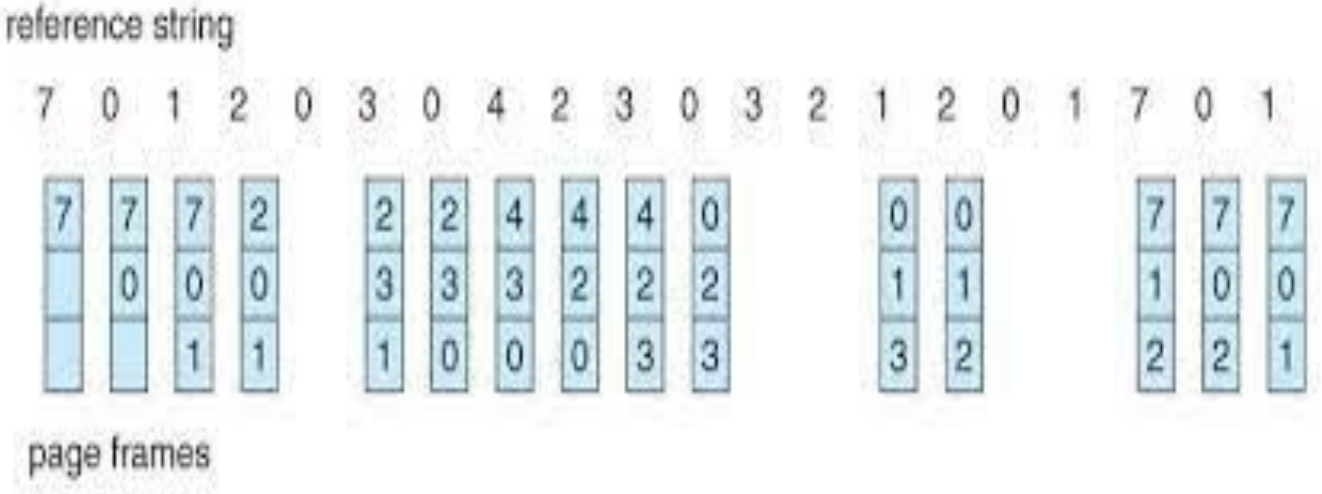
2.Firstly we added the pages until all the frames in the main memory and all of them considered as page fault i.e all the pages of processes are brought from secondary memory which will take more time.

3,Then if the page coming is same as the page present in the first frame then no changes is done and it considered as page hit and next page is compared with same frame page and repeats for all the pages which are present in the list.

4.0therwise it considered as pagefault and page present in first frame is replaced by new page and next page in the set is compared to page present in the second frame if same its considered as page hit otherwise page fault and replaced .

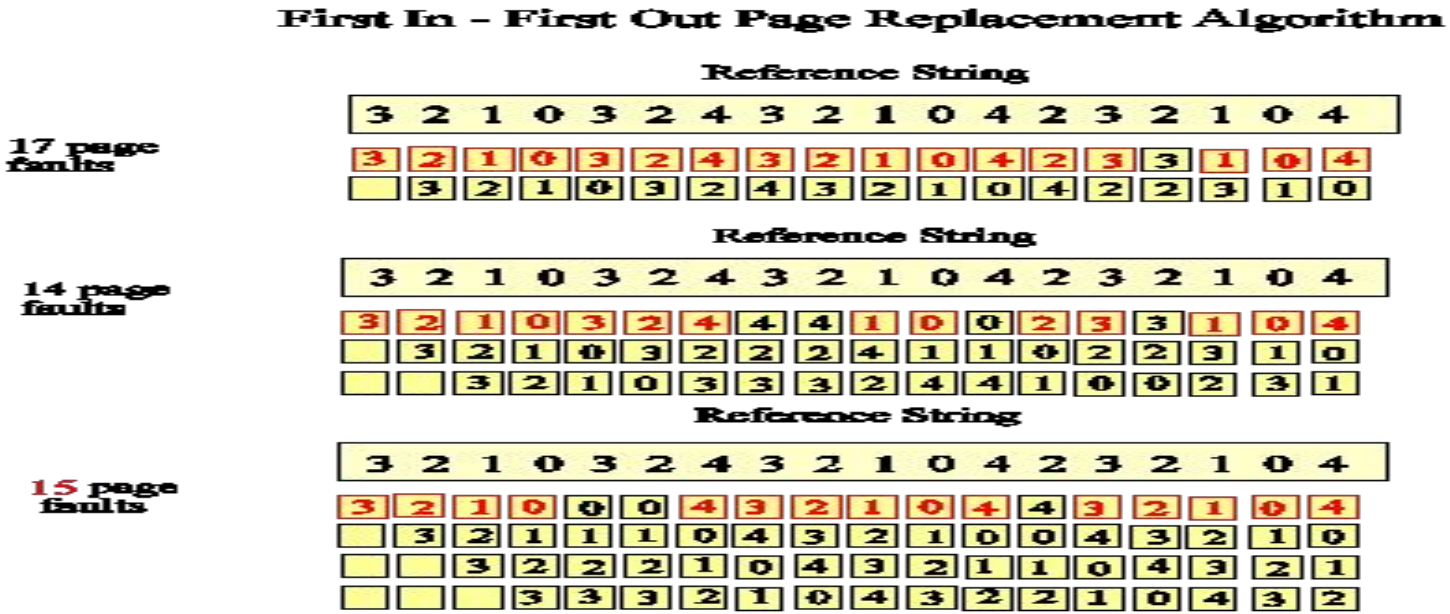
5. Next each pages are compared to increment of the previous frame pages if they are same considered as page hit and new page is compared to same page frame and if its not same its known as page fault and replaced by new page.

Below one is example for FIFO



BELADY'S ANAMOLY

In a FIFO page replacement algorithm, the page that has been in memory the longest is the one selected for replacement when a page fault occurs. The anomaly occurs when increasing the number of page frames (the amount of available memory) leads to an increase in the number of page faults rather than a decrease, which is counterintuitive.



SECOND CHANCE PAGE REPLACEMENT(CLOCK)

The Second-Chance page replacement algorithm, also known as the Clock algorithm, is a modified version of the FIFO (First-In-First-Out) algorithm. It addresses some of the limitations of pure FIFO, particularly the "Belady's Anomaly," by introducing a notion of "second chance" for pages that are candidates for replacement.

Here's how the Second-Chance paging policy works:

1.Page Frame Queue:

- **Maintain a circular queue (a data structure) that represents the order in which pages are brought into memory. Each page in the queue has an additional reference bit associated with it.**

2.Reference Bit:

- **Introduce a reference bit for each page in the page frame set. The reference bit is initially set to 0.**

3.Page Fault Occurs:

- When a page fault occurs (i.e., a program attempts to access a page not currently in physical memory), the operating system examines the page at the front of the queue.

4.Check Reference Bit:

- **If the reference bit of the page at the front is 0, it is a candidate for replacement. The page is replaced, and the new page is added to the rear of the queue.**

5.If Reference Bit is 1 (Second Chance):

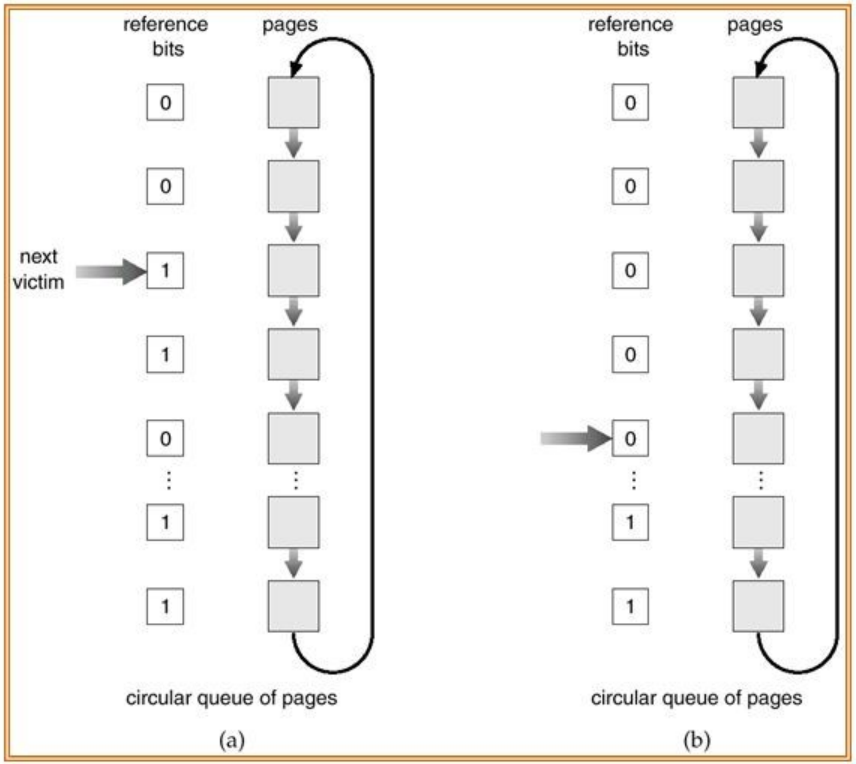
- **If the reference bit is 1, indicating that the page has been referenced recently, the page is given a "second chance." The reference bit is cleared (set to 0), and the page is moved to the rear of the queue. The algorithm then checks the next page in the queue.**

6.Repeat:

- **Steps 3-5 are repeated until a suitable page for replacement is found.**

This algorithm essentially creates a circular queue with an additional reference bit for each page. When a page is considered for replacement, the reference bit is checked. If it's 0, the page is a candidate for replacement. If it's 1, the page is given a "second chance" by moving it to the rear of the queue.

Second-Chance (clock) Page-Replacement Algorithm



THANK YOU