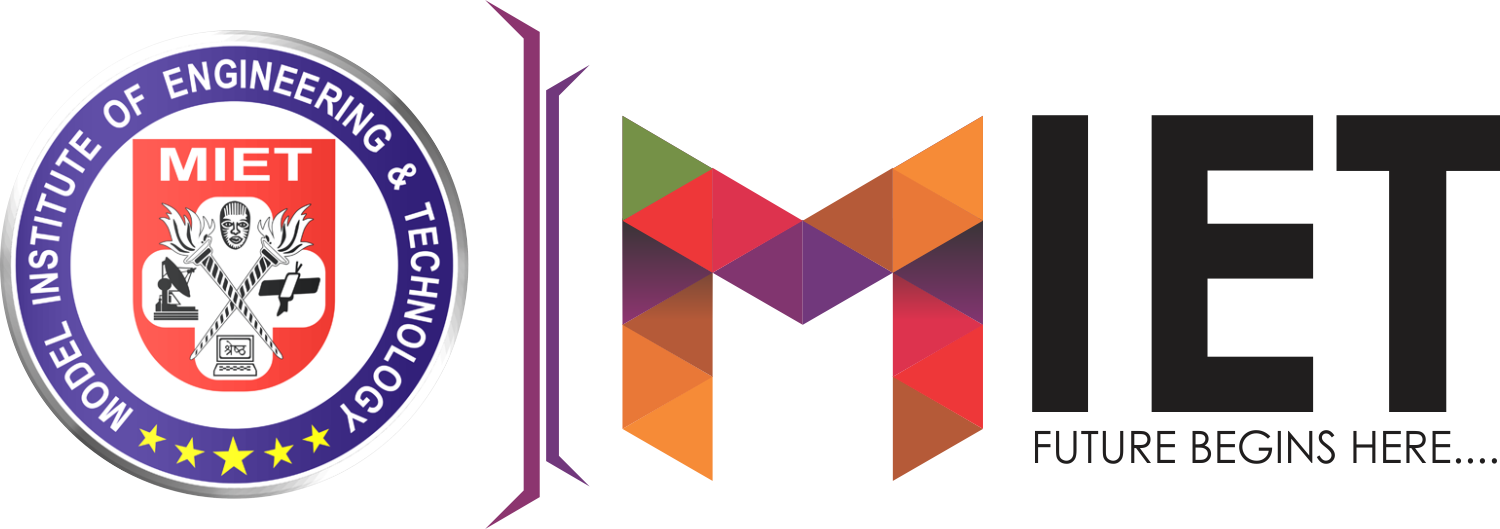
**A Report File on Implementation of the functionality of FIFO, LRU, and OPTIMAL page replacement algorithms.**

**at**

**[CSE, MODEL INSTITUTE OF ENGINEERING AND TECHNOLOGY]**

**BACHELOR OF TECHNOLOGY (Computer Engineering)**

****

**Submitted By**

**Name:** Ayush Thakur, Sonali Tandon

**Roll No:** 2021A1R049, 2021A1R053

**Branch:** CSE

**Semester:** 3rd

[**2021A1R049@mietjammu.in**](mailto:2021A1R049@mietjammu.in) **(Ayush Thakur)**

[**2021A1R053@mietjammu.in**](mailto:2021A1R053@mietjammu.in) **(Sonali Tandon)**

**Linkedin**

[**https://www.linkedin.com/in/ayushthakur-in/**](https://www.linkedin.com/in/ayushthakur-in/)

[**https://www.linkedin.com/in/sonali-tandon-a2842721b/**](https://www.linkedin.com/in/sonali-tandon-a2842721b/)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Abstract**

As the technology has advanced and the program has become more and more complex, the RAM size is not increasing a lot. So typically, when the multi scenario tasks come into picture. There are a lot of apps which are running in the background and when you open up more complex applications, especially Graphic intensive applications they take more RAM and more processing, so this makes the entire ram full and thus memory management comes into picture. Memory management is the process of controlling and coordinating a computer's main memory. Paging is one such methodology in which memory is efficiently managed. In computer operating system, paging is a memory management scheme by which a computer stores and retrieves data from secondary storage for use in main memory. In this game, the operating system. Retrieve data from secondary storage in same size blocks called page. It is actually a part of virtual memory implementations in modern operating system. It helps to prevent external fragmentation. The Memory Management Unit usually divides the virtual address space into pages and therefore is called the Paged Memory Management Unit.

**Keywords**: Page replacement, page fault page hit, page miss, Hit ratio, Paging.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**1. Introduction**

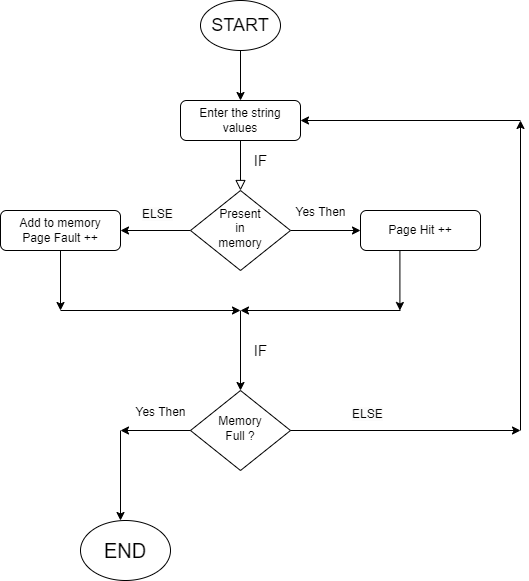
**2. Objective**

Simulator should accept number of physical frames, list of page accesses, and the page replacement algorithm and output the number of faults and whether each access was a fault or not. Implementation of FIFO, LRU, and OPTIMAL page replacement algorithms. There must be an input file (\*.txt) in the directory. The first line in the file should contain the number of physical frames. Each subsequent line represents one page access, and contains exactly one integer, which represents the page number being accessed. All values must be non-negative and fit in the int data type of the system.

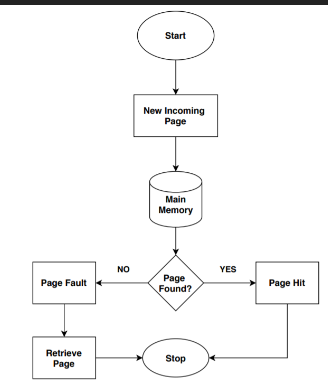
**3. Algorithm and Methodology**

The page replacement algorithm decides which memory page is to be replaced. The process of replacement is sometimes called swap out or write to disk. Page replacement is done when the requested page is not found in the main memory (page fault).There are two main aspects of virtual memory, Frame allocation and Page Replacement. It is very important to have the optimal frame allocation and page replacement algorithm. Frame allocation is all about how many frames are to be allocated to the process while the page replacement is all about determining the page number which needs to be replaced in order to make space for the requested page.

**1.First in First Out (FIFO):**  FIFO algorithm is the simplest of all the page replacement algorithms. In this, we maintain a queue of all the pages that are in the memory currently. The oldest page in the memory is at the front-end of the queue and the most recent page is at the back or rear-end of the queue.



**2.Least Recently Used (LRU):** Least Recently Used algorithm is a page replacement technique used for memory management. According to this method, that page is replaced which is least recently used. Thus, in memory any page that has been unused for a longer period of time than the others are replaced.



**3.Optimal Page Replacement:** **“**Optimal page replacement algorithm" is the most desirable page replacement algorithm that we use for replacing pages. This algorithm replaces the page whose demand in the future is least as compared to other pages from frames (secondary memory). The replacement occurs when the page fault appears. The purpose of this algorithm is to minimize the number of page faults.

**4. Pseudo Code**

**1. FIFO**

#declare variable: values, pages, frames, pageFaults, pageHits

#initialize pageHits=0 and pageFaults=0

Enter the number of pages.

Enter the reference string values.

Enter the number of frames.

Start travesing the string values into the frame.

Check if string value already present in the memory.

If yes (pageHit)

Else (pageFault)

Print pageHits and pageFaults

**2. LRU**

Iterate through the referenced pages.

If the current page is already present in pages:

Remove the current page from pages. Append the current page to the end of pages.

Increment page hits.

Else:

Increment page faults.

If pages contain less pages than its capacity s:

Append current page into pages.

Else:

Remove the first page from pages.

Append the current page at the end of pages.

Return the number of page hits and page faults.

**3. Optimal Page Replacement**