CET-225

Operating Systems

Experiment # 13

Experiment Title

Implementation of Page replacement algorithms: FIFO, LRU, MRU and Optimal algorithm

Assessment of CLO(s): 04

Performed on _____

Student Name:			
Roll No.		**Group**	
Semester		**Session**	

Total (Max)	**Performance (03)**	**Viva (03)**	**File (04)**	**Total (10)**
Marks Obtained				
Remarks (if any)				

Experiment evaluated by

Instructor's Name	**Engr. Bushra Aziz**		
Date		**Signature**	

Objective:

To provide a comprehensive understanding of various page replacement algorithms, including **First-In-First-Out (FIFO)**, **Least Recently Used (LRU)**, **Most Recently Used (MRU)**, and **Optimal Page Replacement**. The manual explains the implementation steps, provides examples, and offers exercises to enhance the practical application of these algorithms. These algorithms are essential for efficient memory management in operating systems, aiming to minimize page faults and optimize system performance.

Theory:
In memory management, **page replacement algorithms** are used to manage pages in memory when a page fault occurs. Four commonly used algorithms are **First-In-First-Out (FIFO)**, **Least Recently Used (LRU)**, **Most Recently Used (MRU)**, and **Optimal**. Below is an implementation manual for these algorithms.

1: First-In-First-Out (FIFO) Algorithm
FIFO works by replacing the oldest page in memory that has been in the queue the longest.

Implementation Steps:
Input the number of frames and the reference string.
Traverse the reference string one page at a time.
If the page is not in memory:
If there is free space in the frames, load the page into memory.
Otherwise, replace the page that has been in memory the longest.
Keep track of page faults.

Example:
Reference String: 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2 Frames: 3

Reference	Memory State	Page Fault
7	[7]	Yes
0	[7, 0]	Yes
1	[7, 0, 1]	Yes
2	[0, 1, 2]	Yes
0	[0, 1, 2]	No
3	[1, 2, 3]	Yes
0	[2, 3, 0]	Yes
4	[3, 0, 4]	Yes
2	[0, 4, 2]	Yes
3	[4, 2, 3]	Yes
0	[2, 3, 0]	Yes
3	[2, 3, 0]	No
2	[2, 3, 0]	No

Total Page Faults: 9

2: Least Recently Used (LRU) Algorithm:

LRU works by replacing the page that has not been used for the longest time.

Implementation Steps:

Input the number of frames and the reference string.
Traverse the reference string one page at a time.
If the page is not in memory:
If there is free space in the frames, load the page into memory.
Otherwise, replace the page that was least recently used.
Keep track of page faults.

Example:

Reference String: 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2 Frames: 3

Reference	Memory State	Page Fault
7	[7]	Yes
0	[7, 0]	Yes
1	[7, 0, 1]	Yes
2	[0, 1, 2]	Yes
0	[0, 1, 2]	No
3	[1, 2, 3]	Yes
0	[2, 3, 0]	Yes
4	[3, 0, 4]	Yes
2	[0, 4, 2]	Yes
3	[4, 2, 3]	Yes
0	[2, 3, 0]	Yes
3	[2, 3, 0]	No
2	[2, 3, 0]	No

Total Page Faults: 10

3: Most Recently Used (MRU) Algorithm

MRU works by replacing the page that was most recently used. It assumes that the most recently used page is less likely to be needed again soon.

Implementation Steps:

Input the number of frames and the reference string.
Traverse the reference string one page at a time.
If the page is not in memory:
If there is free space in the frames, load the page into memory.
Otherwise, replace the page that was most recently used.
Keep track of page faults.
Example:
Reference String: 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2 Frames: 3

Reference	Memory State	Page Fault
7	[7]	Yes
0	[7, 0]	Yes
1	[7, 0, 1]	Yes
2	[0, 1, 2]	Yes
0	[0, 1, 2]	No
3	[1, 2, 3]	Yes
0	[2, 3, 0]	Yes
4	[3, 0, 4]	Yes
2	[0, 4, 2]	Yes
3	[4, 2, 3]	Yes
0	[2, 3, 0]	Yes
3	[2, 3, 0]	No
2	[2, 3, 0]	No

Total Page Faults: 10

4: Optimal Page Replacement Algorithm

The Optimal algorithm works by replacing the page that will not be used for the longest period of time in the future.

Implementation Steps:

Input the number of frames and the reference string.
Traverse the reference string one page at a time.
If the page is not in memory:
If there is free space in the frames, load the page into memory.
Otherwise, look ahead in the reference string and replace the page that is not needed for the longest time.
Keep track of page faults.

Example:

Reference String: 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2 Frames: 3

Reference	Memory State	Page Fault
7	[7]	Yes
0	[7, 0]	Yes
1	[7, 0, 1]	Yes
2	[0, 1, 2]	Yes
0	[0, 1, 2]	No
3	[1, 2, 3]	Yes
0	[2, 3, 0]	Yes
4	[3, 0, 4]	Yes
2	[0, 4, 2]	Yes
3	[4, 2, 3]	Yes
0	[2, 3, 0]	Yes
3	[2, 3, 0]	No
2	[2, 3, 0]	No

Total Page Faults: 7

Exercises:

Write a Python program to implement the FIFO page replacement algorithm.
Write a Python program to implement the LRU page replacement algorithm.
Write a Python program to implement the MRU page replacement algorithm.
Write a Python program to implement the optimal page replacement algorithm.
Compare the performance of all algorithms for varying frame sizes and reference strings.