

Assignment 4: Analyzing and Implementing the LRU Page Replacement Algorithm

Objective: The goal of this assignment is to help students understand the Least Recently Used (LRU) page replacement algorithm through theoretical analysis, practical exercises, and critical reflection.

T-1: Consider a demand-paging system in which a process is allocated three memory frames. When a page fault occurs in the process, let p_1 , p_2 and p_3 be the three pages of the process that are resident in memory. Let the timestamps of arrival (i.e., the time at which the page was loaded most recently) for these three pages be a_1 , a_2 and a_3 respectively. Let the timestamps of last reference to these pages be r_1 , r_2 and r_3 respectively.

Let the Least Recently Used (LRU) page-replacement scheme be employed by the demand-paging system. If p_2 is the page that is chosen to be replaced as part of servicing the page fault, determine the relationships among a_1 , a_2 , a_3 , r_1 , r_2 , and r_3 . Based on these relationships, select from the following list the statement that is CERTAIN to be TRUE and explain why.

- A. r_1 is earlier than r_3
- B. r_1 is later than r_2
- C. r_3 is earlier than r_1
- D. r_2 is later than r_3
- E. r_2 is earlier than r_3

T-2: Consider the following page reference string:

7, 2, 3, 1, 2, 5, 3, 4, 6, 7, 7, 1, 0, 5, 4, 6, 2, 3, 0, 1

Assuming demand paging with **three frames**.

Simulate (write `LRU.c` program) the LRU page replacement process using given sequence and frame allocation while keeping track of:

- The contents of the frames at each step
- The page fault occurrences
- Which page is replaced during each fault

➤ Remember that all frames are initially empty, so your first unique pages will cost one fault each.

Briefly explain the working of your program.

Expected Output (Truncated):

Page	Frames	Fault
7	7 - -	Page Fault
2	7 2 -	Page Fault
3	7 2 3	Page Fault
	...	
Total Page Faults: X		

Submission Guidelines

- Submit your `LRU.c` source files for LRU page replacement algorithm via GitHub.
- Include a README file explaining how to run your program and any assumptions made. Also, provide a separate document with your answers to the posted questions.
- Ensure your code compiles and runs without errors before submission.
- DUE DATE: **20th November** (Wednesday) by the end of day.