Programming Assignment #4: Page Replacement Algorithms

[10 points]

Objective: This assignment aims to understand and implement different page replacement algorithms in a demand-paging scenario and analyze their performance in terms of the number of page faults.

Requirements:

1. Implement the following page replacement algorithms: LRU (Least Recently Used), Optimal, and FIFO (First In First Out)
2. Create a mechanism to accept the given page reference string and the specified number of frames (4 frames in this case) as input to simulate page replacement.
3. Develop algorithms to simulate page replacement and calculate the number of page faults that occur during the simulation for each implemented algorithm.
4. For each algorithm, provide a clear and structured output displaying the number of page faults that occurred at each step of the simulation.

Implementation: Write a script that meets the abovementioned requirements.

Sample Input:

Page reference string: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5
Number of frames: 4

Sample Output:

For LRU Algorithm:

- Step 1: Page fault (1) - Page Table: {1}, Frames: [1], Faults: 1
- Step 2: Page fault (2) - Page Table: {1, 2}, Frames: [1, 2], Faults: 2
- Step 3: Page fault (3) - Page Table: {1, 2, 3}, Frames: [1, 2, 3], Faults: 3
- ... (continue until end of page reference string)
- Total Page Faults: [Total Faults]

For Optimal Algorithm:

- Step 1: Page fault (1) - Page Table: {1}, Frames: [1], Faults: 1
- Step 2: Page fault (2) - Page Table: {1, 2}, Frames: [1, 2], Faults: 2
- Step 3: Page fault (3) - Page Table: {1, 2, 3}, Frames: [1, 2, 3], Faults: 3
- ... (continue until end of page reference string)
- Total Page Faults: [Total Faults]

For FIFO Algorithm:

- Step 1: Page fault (1) - Page Table: {1}, Frames: [1], Faults: 1
- Step 2: Page fault (2) - Page Table: {1, 2}, Frames: [1, 2], Faults: 2
- Step 3: Page fault (3) - Page Table: {1, 2, 3}, Frames: [1, 2, 3], Faults: 3
- ... (continue until end of page reference string)
- Total Page Faults: [Total Faults]

Submission Guidelines:

Please ensure that your assignment submission includes well-documented source code accompanied by a README file detailing the program's functionality and providing instructions for execution. Once the assignment is complete, upload the code to Git, and share the repository link. Additionally, please attach a zip file containing the entire project.

Grading breakdown:

1. **Functionality:**
    - Evaluate the correctness of the implemented page replacement algorithms. Ensure that the algorithms accurately simulate page replacement and calculate the correct number of page faults for the given input.
    - Assess the efficiency of the algorithms in terms of time complexity and resource usage. Consider factors such as the number of comparisons, memory accesses, and overall execution time.
    - Check if the program fulfills all functionality requirements, including proper input handling, accurate page fault calculation, and clear output display. Ensure that the program covers all aspects of the assignment description.
    - Evaluate the program's robustness in handling unexpected inputs or scenarios gracefully. Ensure that the program provides appropriate error messages and does not crash or encounter unexpected behavior.
2. **Script Submission:**
    - Proper submission of the script via the designated platform.
    - Adequate documentation or comments explaining the implementation choices.