

**Laboratory Exercise No. 6**

Submit the following:

- ✓ Your source code.  
Java files provided in submission
  
- ✓ A short report (1-2 paragraphs) explaining your observations on the impact of frame size on page fault count.

In the FIFO page replacement algorithm, frame size is one of the important factors that determine the number of page faults faced during memory management. In fact, with an increased frame size, the system can hold a higher number of pages at any given time, and generally, this results in the reduction of page faults. This is because an increased frame size allows the usage of more pages to be kept in the memory, thus the number of page swaps-in and reloads that would be incurred is reduced. For instance, with a frame size of 3, the algorithm yields more page faults compared to that of a frame size of 5, where the probability of getting more requested pages to be resident in the memory is high.

However, bigger frames are not entirely successful as size grows: at some point the greatest gain reduces off as pages in the working set of pages outnumber frame capacity. Then, though more and more frames continue to drive down page fault rate, the gain may not necessarily be proportional to the increment in frame size.

- ✓ Screenshots or logs showing your program's output for different frame sizes.

```
Enter page requests (comma-separated): 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 3
Enter number of frames (3-5): 3
Step 1: [7]
Step 2: [7, 0]
Step 3: [7, 0, 1]
Step 4: [0, 1, 2]
Step 5: [0, 1, 2]
Step 6: [1, 2, 3]
Step 7: [2, 3, 0]
Step 8: [3, 0, 4]
Step 9: [0, 4, 2]
Step 10: [4, 2, 3]
Step 11: [2, 3, 0]
Step 12: [2, 3, 0]
Step 13: [2, 3, 0]
Step 14: [2, 3, 0]
Total page faults: 10
Final state of frames: [2, 3, 0]
```

```
Enter page requests (comma-separated): 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 3
Enter number of frames (3-5): 4
Step 1: [7]
Step 2: [7, 0]
Step 3: [7, 0, 1]
Step 4: [7, 0, 1, 2]
Step 5: [7, 0, 1, 2]
Step 6: [0, 1, 2, 3]
Step 7: [0, 1, 2, 3]
Step 8: [1, 2, 3, 4]
Step 9: [1, 2, 3, 4]
Step 10: [1, 2, 3, 4]
Step 11: [2, 3, 4, 0]
Step 12: [2, 3, 4, 0]
Step 13: [2, 3, 4, 0]
Step 14: [2, 3, 4, 0]
Total page faults: 7
Final state of frames: [2, 3, 4, 0]
```

```
Enter page requests (comma-separated): 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 3
Enter number of frames (3-5): 5
Step 1: [7]
Step 2: [7, 0]
Step 3: [7, 0, 1]
Step 4: [7, 0, 1, 2]
Step 5: [7, 0, 1, 2]
Step 6: [7, 0, 1, 2, 3]
Step 7: [7, 0, 1, 2, 3]
Step 8: [0, 1, 2, 3, 4]
Step 9: [0, 1, 2, 3, 4]
Step 10: [0, 1, 2, 3, 4]
Step 11: [0, 1, 2, 3, 4]
Step 12: [0, 1, 2, 3, 4]
Step 13: [0, 1, 2, 3, 4]
Step 14: [0, 1, 2, 3, 4]
Total page faults: 6
Final state of frames: [0, 1, 2, 3, 4]
```