Assignment D

1 Problem Statement

Write a java program (using OOP features) to implement paging simulation using
1. Least Recently Used(LRU) 2.Optimal algorithm

2 Learning Objectives

In this assignment, students will :

1. Understand the concept of page replacement in operating system.

3 Learning Outcomes

After completion of this assignment, students will be able to paging simulation using:

1. Least Recently Used(LRU)

2. Optimal algorithm

4 Requirements

Hardware : 64-bit 2.8 GHz processor, 4 GB RAM

Software : 64-bit OS, Web Browser

5 Theory

In operating systems that use paging for memory management, page replacement algorithm are needed to decide which page needed to be replaced when new page comes in. Whenever a new page is referred and not present in memory, page fault occurs and Operating System replaces one of the existing pages with newly needed page. Different page replacement algorithms suggest different ways to decide which page to replace. The target for all algorithms is to reduce number of page faults.

In Least Recently Used (LRU) algorithm is a Greedy algorithm where the page to be replaced is least recently used. The idea is based on locality of reference, the least recently used page is not likely

Algorithm for LRU:-

Let capacity be the number of pages that memory can hold. Let set be the current set of pages in memory.

1- Start traversing the pages.

i) If set holds less pages than capacity.

a) Insert page into the set one by one until the size of set reaches capacity or all page requests are processed.

b) Simultaneously maintain the recent occurred index of each page in a map called indexes.

c) Increment page fault

ii) Else If current page is present in set, do nothing. Else

a) Find the page in the set that was least recently used. We find it using index array. We basically need to replace the page with minimum index.

b) Replace the found page with current page.

c) Increment page faults.

d) Update index of current page.

2. Return page faults.

Algorithm for Optimal Algorithm

The idea is simple, for every reference we do following :

1.If referred page is already present, increment hit count.

2.If not present, find if a page that is never referenced in future. If such a page exists, replace this page with new page. If no such page exists, find a page that is referenced farthest in future. Replace this page with new page

6 Output

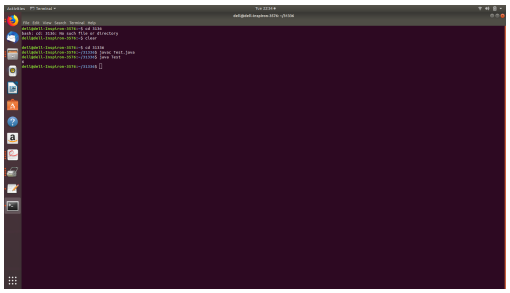

Figure 1: LRU

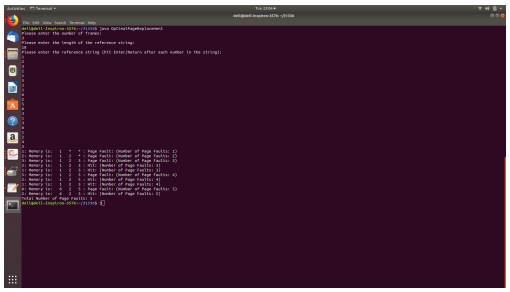

Figure 2

7 Conclusion

Hence, we learnt implementation of page replacement algorithms in operating system.